2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DTIC

AD-A236 942

THESIS

ANALYSIS AND SYNTHESIS OF RADIATIVE
HEAT TRANSFER IN LONGITUDINAL FINS
IN FREE SPACE AND NON-FREE SPACE

by

Dennis R. Johnson

June 1990

Thesis Advisor: Prof. Allan D. Kraus

Approved for public release; distribution unlimited

91-01820

91 6 11 099

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified

1b Restrictive Markings

2a Security Classification Authority

3 Distribution Availability of Report
Approved for public release; distribution is unlimited.

2b Declassification/Downgrading Schedule

4 Performing Organization Report Number(s)

5 Monitoring Organization Report Number(s)

6a Name of Performing Organization
Naval Postgraduate School

6b Office Symbol *(If Applicable)* 39

7a Name of Monitoring Organization
Naval Postgraduate School

6c Address *(city, state, and ZIP code)*
Monterey, CA 93943-5000

7b Address *(city, state, and ZIP code)*
Monterey, CA 93943-5000

8a Name of Funding/Sponsoring Organization

8b Office Symbol *(If Applicable)*

9 Procurement Instrument Identification Number

8c Address *(city, state, and ZIP code)*

10 Source of Funding Numbers

Program Element Number	Project No	Task No	Work Unit Accession No

11 Title *(Include Security Classification)* ANALYSIS AND SYNTHESIS OF RADIATIVE HEAT TRANSFER IN LONGITUDINAL FINS IN FREE SPACE AND NON-FREE SPACE

12 Personal Author(s) Dennis R. Johnson

13a Type of Report
Master's Thesis

13b Time Covered
From To

14 Date of Report *(year, month,day)*
1990 June

15 Page Count
158

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes

Field	Group	Subgroup

18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Radiative Heat Transfer, Longitudinal Fins, Free Space and Non-Free Space

19 Abstract *(continue on reverse if necessary and identify by block number*

The objective of this thesis is to develop an interactive computer program that will analyze and synthesize radiative heat transfer in longitudinal fins. The analysis procedure determines the amount of heat transferred from the fin given the fin base temperature, fin dimensions and thermal properties. The synthesis procedure is the converse problem: it determines the size of the fin required to dissipate a specified amount of heat given the thermal characteristics of the fin. In addition, the program is capable of performing the analysis/synthesis of three fin profiles (rectangular, trapezoidal and triangular) in two environments (free space and non-free space). Free space is considered as the absence of external heat sources or interception of the heat dissipated by the fin whereas non-free space includes the effect of external heat sources and neighboring structures. A theoretical analysis of heat transfer from radiating longitudinal fins will be presented along with a user oriented computer program. Finally, detailed examples will be provided to illustrate the different types of problems, profiles and environments.

20 Distribution/Availability of Abstract
[X] unclassified/unlimited [ ] same as report [ ] DTIC users

21 Abstract Security Classification
Unclassified

22a Name of Responsible Individual
Allan D. Kraus

22b Telephone *(Include Area code)*
(408)-646-2730

22c Office Symbol
EC/Ks

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted security classification of this page

All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited.

ANALYSIS AND SYNTHESIS OF RADIATIVE HEAT TRANSFER IN LONGITUDINAL FINS IN FREE SPACE AND NON-FREE SPACE

by

Dennis R. Johnson
Lieutenant Commander, United States Navy
B.S., North Carolina State University, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1990

Author: ______________________
Dennis R. Johnson

Approved By: ______________________
Allan D. Kraus, Thesis Advisor

____
Sue Brown, Second Reader

____
E. Roberts Wood, Chairman, Department of Aeronautical and Astronautical Engineering

ABSTRACT

The objective of this thesis is to develop an interactive computer program that will analyze and synthesize radiative heat transfer in longitudinal fins. The analysis procedure determines the amount of heat transferred from the fin given the fin base temperature, fin dimensions and thermal properties. The synthesis procedure is the converse problem: it determines the size of the fin required to dissipate a specified amount of heat given the thermal characteristics of the fin. In addition, the program is capable of performing the analysis/synthesis of three fin profiles (rectangular, trapezoidal and triangular) in two environments (free space and non-free space). Free space is considered as the absence of external heat sources or interception of the heat dissipated by the fin whereas non-free space includes the effect of external heat sources and neighboring structures. A theoretical analysis of heat transfer from radiating longitudinal fins will be presented along with a user oriented computer program. Finally, detailed examples will be provided to illustrate the different types of problems, profiles and environments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Q = heat flow (W)

Q_I = ideal heat dissipation (W)

E = external heat input (W)

x = height coordinate along the fin (m)

dx = differential element along x (m)

A = area normal to the direction of heat flow(m^2)

L = length of fin (m)

b = height of fin (m)

δ = thickness of fin (m)

δ_0 = thickness at fin base (m)

δ_E = thickness at fin tip (m)

$\delta(x)$ = thickness of fin at x (m)

k = thermal conductivity of fin material ($W/m\text{-}^{\circ}K$)

α = surface absorptivity ($0 \leq \alpha \leq 1$) (dimensionless)

ϵ = surface emissivity ($0 \leq \epsilon \leq 1$) (dimensionless)

σ = Stefan-Boltzman constant (5.6697×10^{-8} $W/m^2\text{-}K^4$)

$T(x)$ = fin temperature at x ($^{\circ}K$)

$\frac{dT}{dx}$ = temperature gradient in the x direction ($^{\circ}K/m$)

T_0 = temperature at the fin base ($^{\circ}K$)

T_S = temperature of the environment ($T_S = 0$ $^{\circ}K$ in space)

T_E = temperature at the fin tip ($^{\circ}K$)

K_1 = constant = $2 \sigma \epsilon$ ($W/m^2\text{-}^{\circ}K^4$)

K_2 = constant = $E \alpha$ (W/m^2)

$\cap$ = fin efficiency (dimensionless)

I. INTRODUCTION

A. DEFINITIONS

The American Heritage Dictionary of the English Language defines heat as " a form of energy associated with the motion of atoms or molecules in solids and capable of being transmitted through solid and fluid media by conduction, through fluid media by convection, and through empty space by radiation " [Ref. 1: p. 608]. The transfer of heat within a homogeneous substance because of a temperature differential is called conduction. Convection is the transfer of heat between a solid body and a contiguous, moving fluid at a different temperature. Radiation is the transfer of heat by electromagnetic waves in a medium or a vacuum.

The American Heritage Dictionary of the English Language defines a fin as "a projecting vane used for cooling, as on a radiator or engine cylinder" [Ref. 1: p. 492]. The use of fins is a commonly used technique to increase exposed surface area and hence increase the rate of heat transfer from a body to its surroundings.

B. EXAMPLES OF HEAT TRANSFER BY FINS

Applications of heat transfer augmentation through the use of fins range from motorcycle engine blocks to space

stations. A motorcycle's internal combustion gasoline engine generates a large amount of heat that must be dissipated in order for the engine to remain within the permissible temperature limits of the cylinder liners. The engine casing does not provide sufficient surface area to dissipate this heat; therefore, external fins are installed to increase the heat transfer as indicated in Figures 1A and 1B [Ref. 2: pp. 1,4,56]. Similarly the Space Station, "Freedom", will require hundreds of square meters of surface area to dissipate the heat generated by its electronic equipment as shown in Figure 2 [Ref. 3: p. 29].

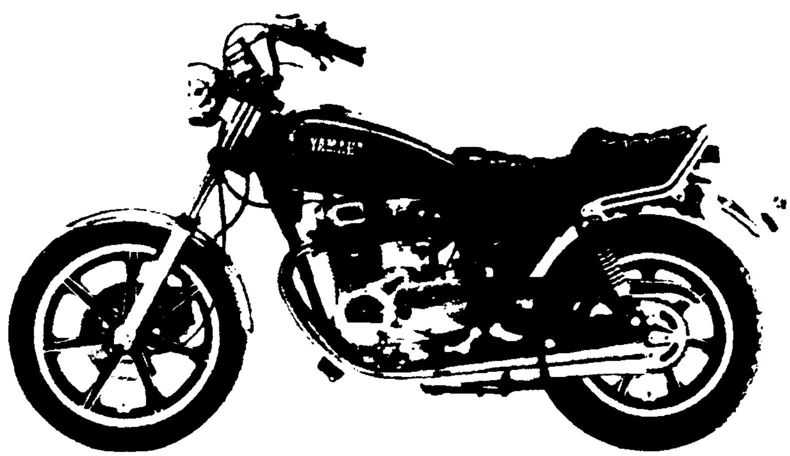

Figure 1A Fins on Motorcycle Engine

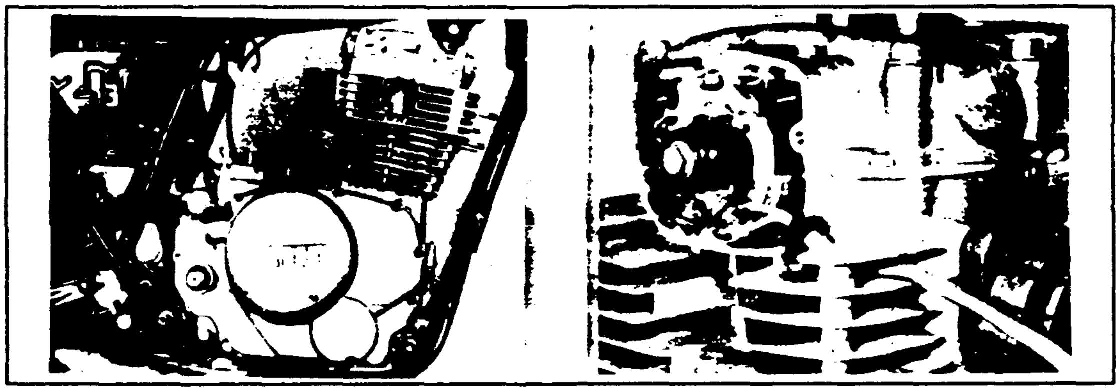

Figure 1B Fins on Motorcycle Engine

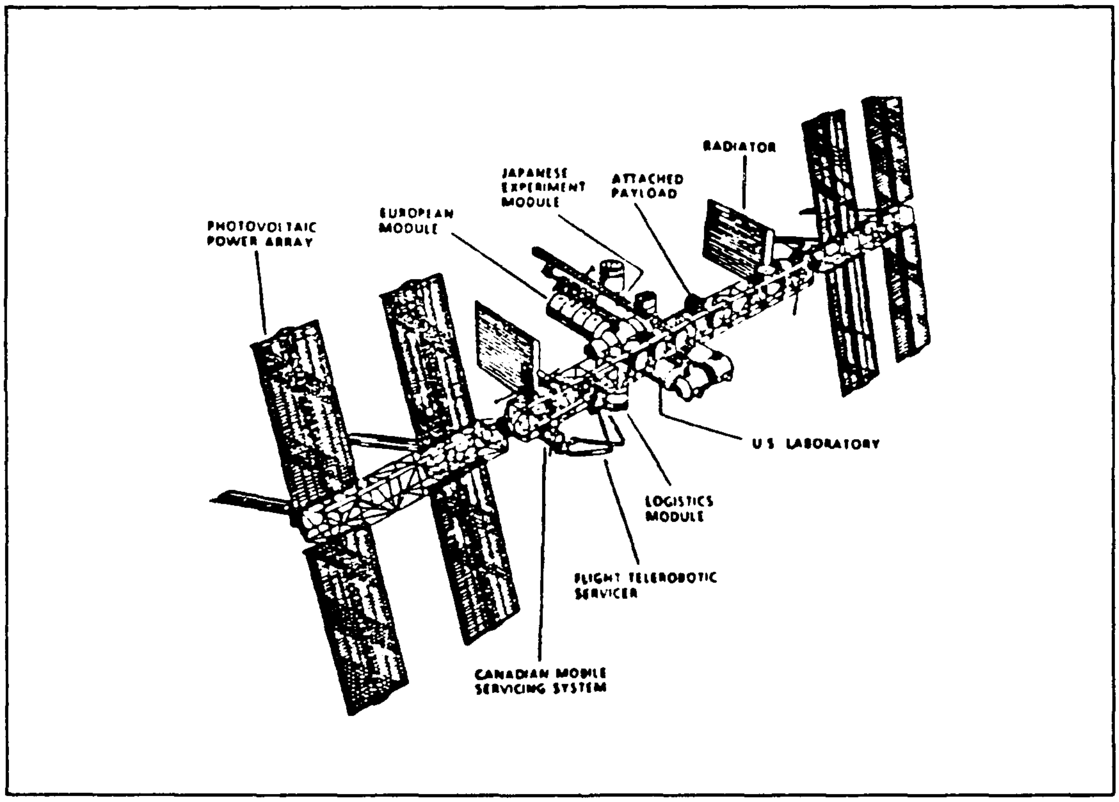

Figure 2 Radiators on Space Station

C. PROFILES

Fins are fabricated in many sizes and shapes [Ref. 4: pp. 102 - 103], [Ref. 5: pp. 3-22]; however, for this

thesis only three common shapes of longitudinal fins are investigated. These are the fins of rectangular, trapezoidal and triangular profile. Longitudinal in the foregoing sense means "placed or running lengthwise" [Ref. 1: p. 769]; that is to say, the analysis is simplified to a one dimensional variation of temperature defined along the "x" axis and a uniform distribution of temperature along the other two axes. Figure 3 shows these profiles.

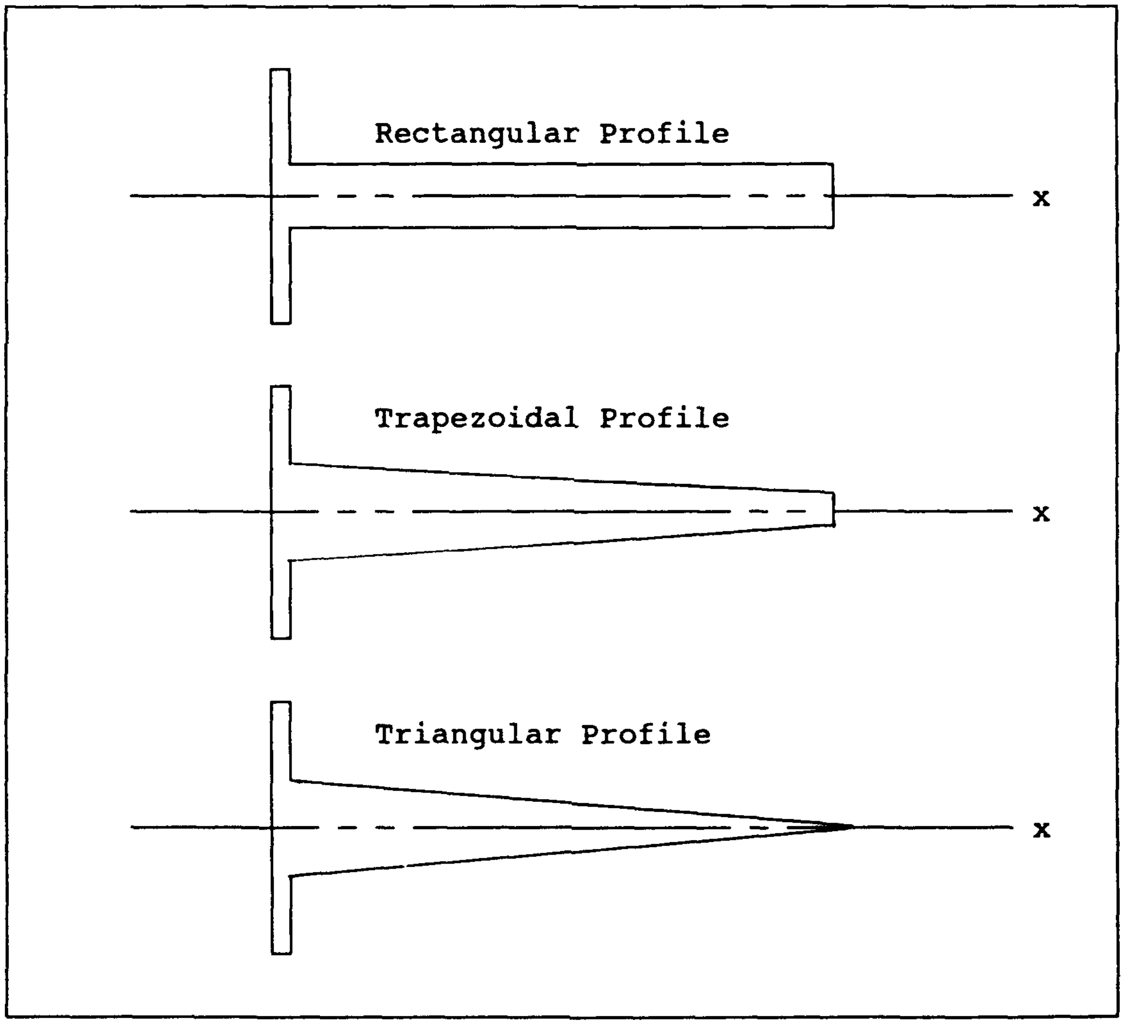

Figure 3 Longitudinal Fin Profiles

D. ENVIRONMENTS

The three types of fins will be analyzed with respect to two environments: free space and non-free space. Free space in this sense means free of external heat sources as seen by the fin or radiator or free of nearby surfaces that intercept radiating energy from the fin . Non-free space includes the effect of external heat sources. For example, a fin exposed to sunlight will have an external solar heat flux incident upon it in addition to the heat conducted through the base of the fin. A fin in the shade will have no such external source and therefore may radiate into "free space". For the non-free space environment, the term "external heat" will also include the following:

1. Radiative heat incident on the fin from external heat sources.
2. Radiative heat from the fin intercepted by nearby surfaces such as solar panels or other dissipating fins or the vehicle structure itself.

The radiative transfer between a fin and its environment can be quite complex depending on the number of heat sources, the geometric shapes of the heat sources and their spacial arrangements (view factors) [Ref. 6: pp. 1 - 7].

An idea of the free space and non-free space environments is shown in Figure 4.

E. PROBLEM TYPES

Two types of problems will be investigated. Analysis is the problem of determining the amount of heat transferred to

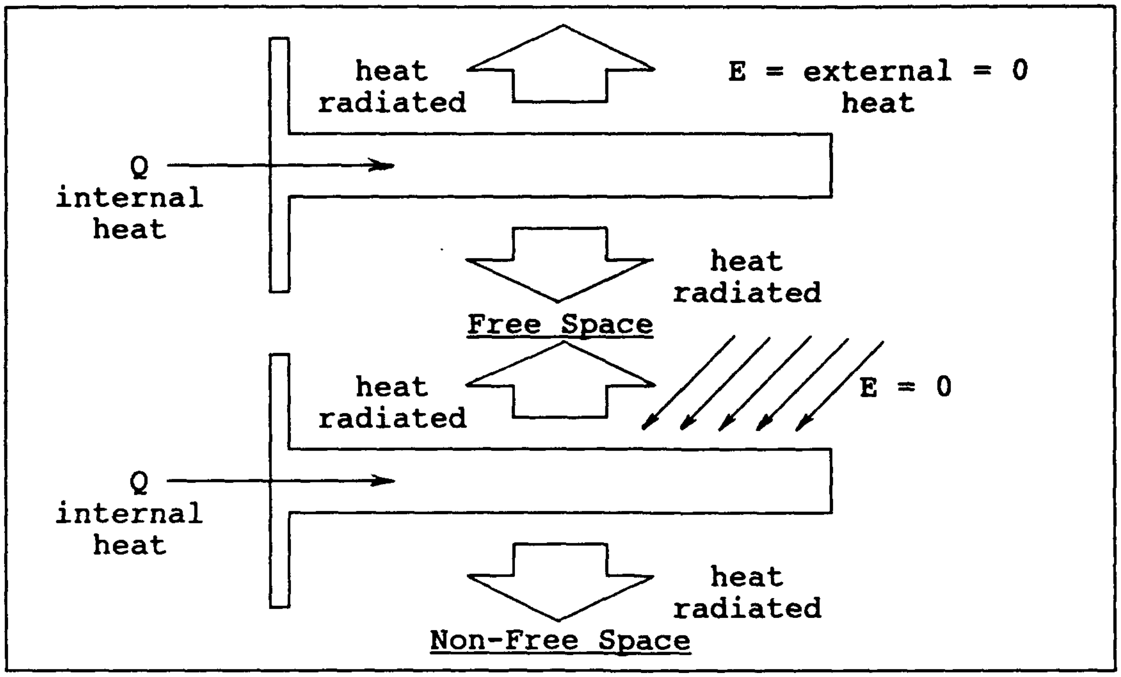

Figure 4 Free Space Vs Non-Free Space Environments

the environment from a fin with given dimensions, base temperature, external heat sources nearby and particular thermal properties. Synthesis is the converse problem: given a specified amount of heat to be dissipated at a certain base temperature and with certain thermal properties, the fin is sized so as to efficiently transmit this heat to the environment. These input/output relationships are shown in Figure 5.

F. STEADY-STATE VS NON-STEADY-STATE

Only a steady-state is considered. Within this static context, the temperature does not change with time. Mathematically this means that any derivatives of

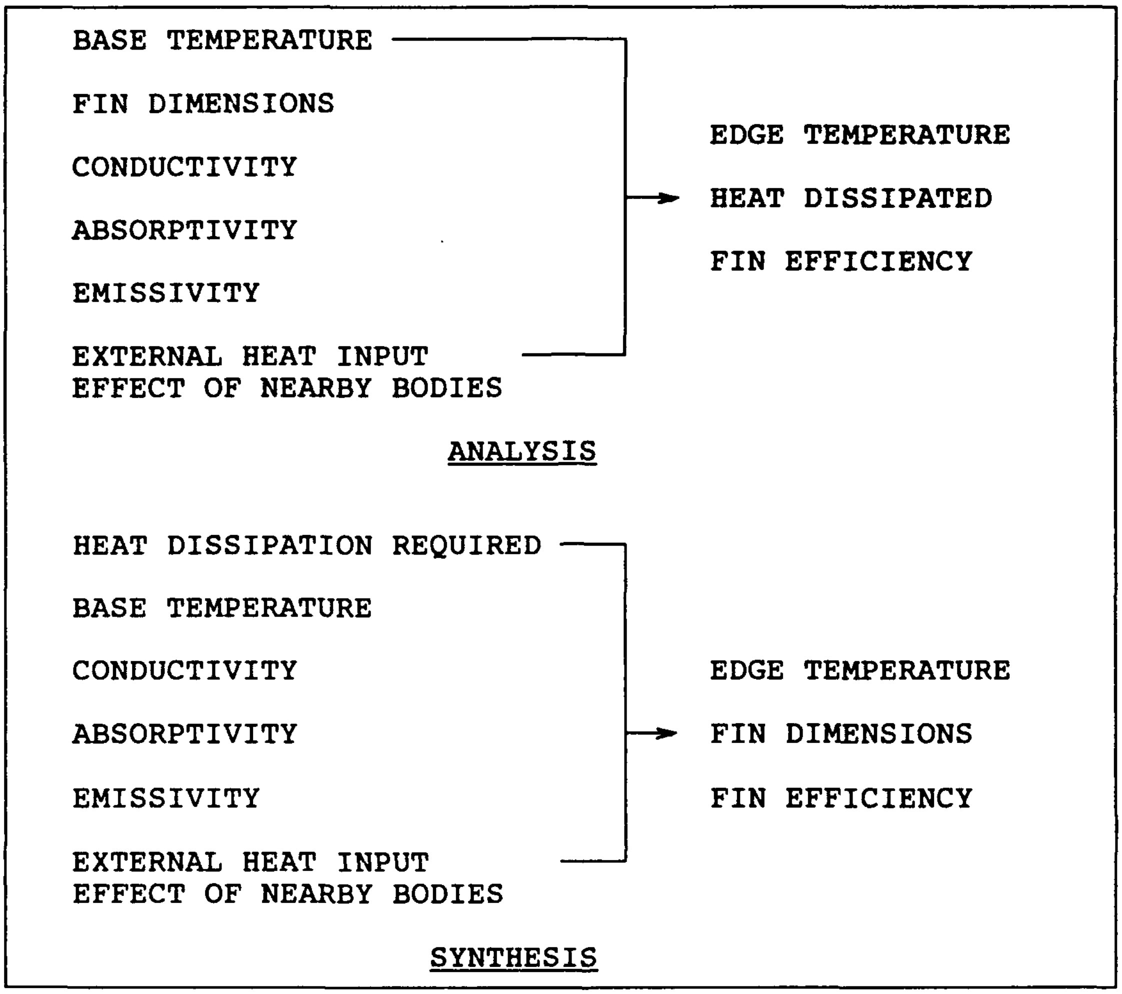

Figure 5 Analysis vs Synthesis Problems

temperature with respect to time are zero. This is in contrast to the non-steady-state or dynamic situation where temperature is a function of time. A satellite emerging from the Earth's shadow would experience a warming period of increasing temperature (non-steady-state) until it reaches thermal equilibrium with respect to the space environment at which time, the temperature would remain constant (steady-state).

G. OVERVIEW

In summary, this thesis will analyze longitudinal fins of three basic profiles (rectangular, trapezoidal and triangular) in two environments (free space and non-free space) to obtain the solution to two types of problems (analysis and synthesis). After a theoretical analysis is developed, a computer program will be presented that will treat these situations through an interactive series of user oriented menus. Finally, detailed examples of typical applications will be presented. An overview is shown in Figure 6.

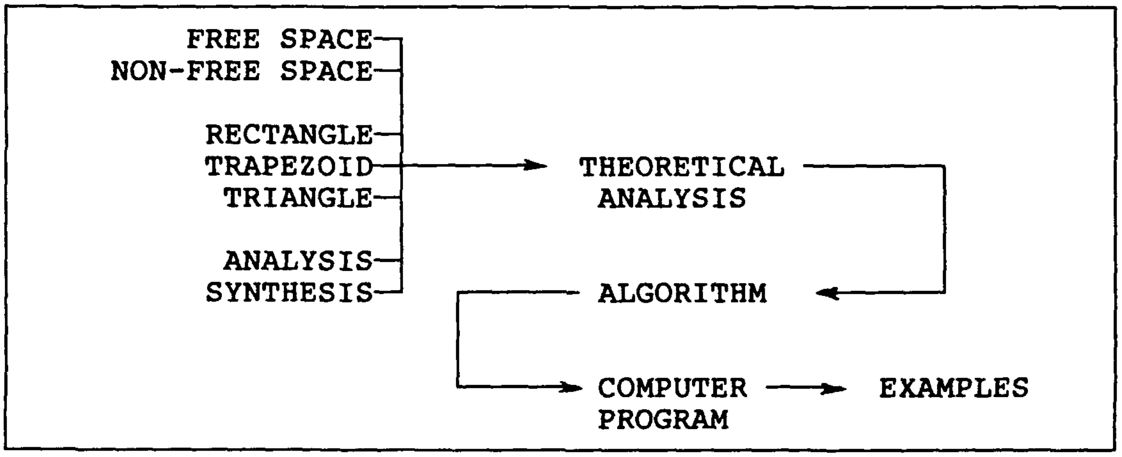

Figure 6 Overview

II. ANALYSIS OF RADIATIVE HEAT TRANSFER IN LONGITUDINAL FINS

A. PRELIMINARY

1. Restrictions

The analysis will be restricted to an environment where the absence of a fluid medium (vacuum) precludes heat transfer by convection. Here, only the non-free case will be analyzed for each of the profiles (rectangular, trapezoidal, triangular) and the two types problems (analysis, synthesis) since it is the more general case and includes the free space case.

2. General Heat Balance Equation

The general heat balance equation as derived from the principle of the conservation of energy states that the amount of heat gained by an object from its surroundings is equal to the heat lost plus any heat stored by the object.

$$\text{HEAT GAINED} = \text{HEAT LOST} + \text{HEAT STORED} \qquad [1]$$

3. Heat Transfer by Conduction

The transfer of heat within a homogeneous substance because of a temperature differential is called conduction. The simplest case of one dimensional heat flow by conduction in a solid is shown in Figure 7.

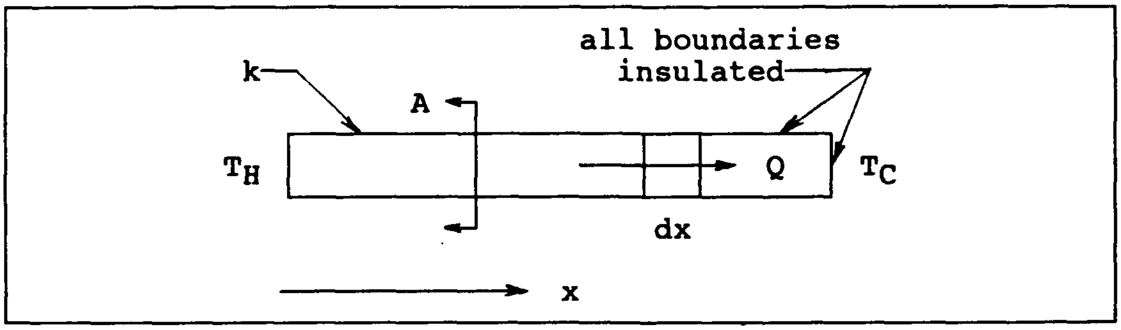

Figure 7 Heat Transfer by Conduction

where:

Q = heat flow (W)

k = thermal conductivity which is a property of the material (W/m-^{o}K)

x = length coordinate over which the energy is transferred (m)

A = area normal to the direction of heat flow (m^2)

dx = differential element along x (m)

$\frac{dT}{dx}$ = the temperature gradient in the x direction (^{o}K/m)

T_H = temperature of "hot" end (^{o}K)

T_C = temperature of "cold" end (^{o}K)

The transfer of energy by conduction is described by Fourier's law [Ref. 7: p. 26]:

$$Q = -kA\frac{dT}{dx} \qquad [2]$$

Equation [2] states that the amount of energy transferred is proportional to the temperature differential along x and the cross-sectional area normal to the direction

of heat flow. The constant of proportionality, k, is called the thermal conductivity and depends on the material. The minus sign conforms to the convention that heat flows from a hot source to a cold sink and assures a positive heat flow in the presence of a negative temperature gradient. More complex equations involving two and three dimensions, steady-state and transient heat flow in different coordinate systems (rectangular, cylindrical and spherical) can be derived [Ref. 7: p. 17] (the one dimensional case is a considerable simplification) but will not be given here. The thermal conductivities of some spacecraft materials are given in Table I [Ref. 8: p. 269].

TABLE I THERMAL CONDUCTIVITY

Material (at 298 $^{\circ}$K)	k (W/m-$^{\circ}$K)
Aluminum	210
Aluminum alloys	117 - 175
Magnesium	157
Magnesium alloys	52 - 111
Titanium	21
Stainless steel	16.2
Teflon	0.25

4. Heat Transfer by Radiation

Radiation is the transfer of heat by electromagnetic waves in a medium or a vacuum. A simple example is shown in Figure 8.

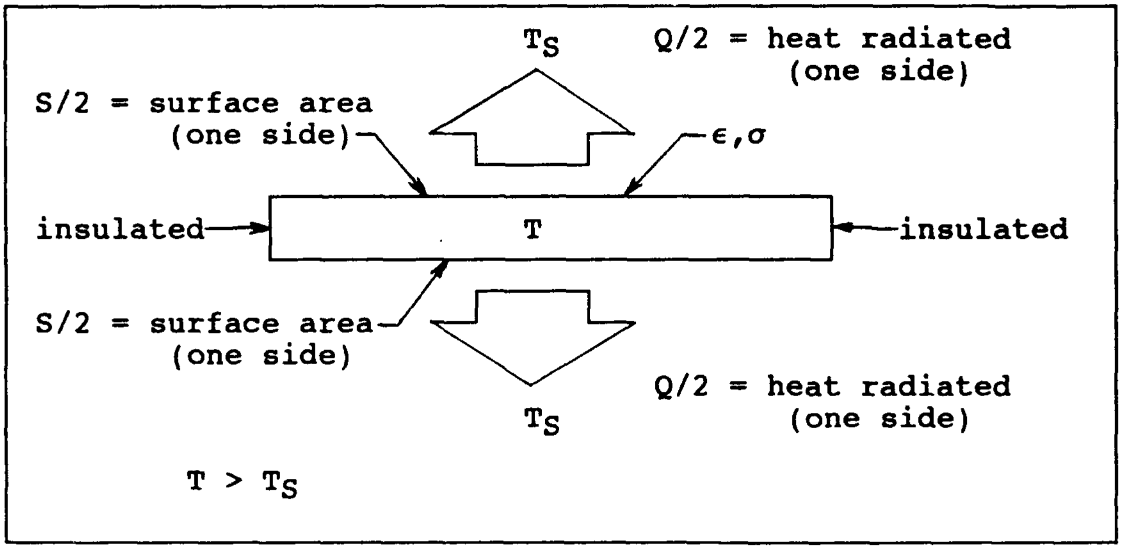

Figure 8 Heat Transfer by Radiation

where:

Q = amount of heat radiated (W)

ϵ = emissivity of the surface ($0 \le \epsilon \le 1$) (dimensionless)

σ = Stefan-Boltzman constant (5.6697×10^{-8} W/m^2-K^4)

S = surface area for energy transfer (m^2)

T = temperature of the object (^{o}K)

T_S = temperature of the environment (^{o}K)

The transfer of energy by radiation (to free space at T = 0 ^{o}K) is described by Stefan-Boltzman Law [Ref. 9: p. 18]:

$$Q = \sigma \epsilon S T^4 \qquad [3]$$

Note the presence of the highly nonlinear T^4 term. The emissivity of the surface is determined by the type of material. Table II lists the emissivities of some typical spacecraft materials [Ref. 8: p. 275].

TABLE II EMISSIVITY

Material (9.3μm, 310 ^{o}K)	Emissivity
Black paint	0.90
White paint	0.90
Graphite/epoxy	0.85
Lampblack	0.84
Solar cells	0.82
Optical solar reflector	0.80
Anodized aluminum	0.80
Aluminized kapton	0.60
Tiodized titanium	0.60
Aluminum	0.06
Gold	0.03

B. RECTANGULAR FIN

1. Geometry

The geometry, terminology and coordinate system for the longitudinal fin of rectangular profile are shown in Figure 9:

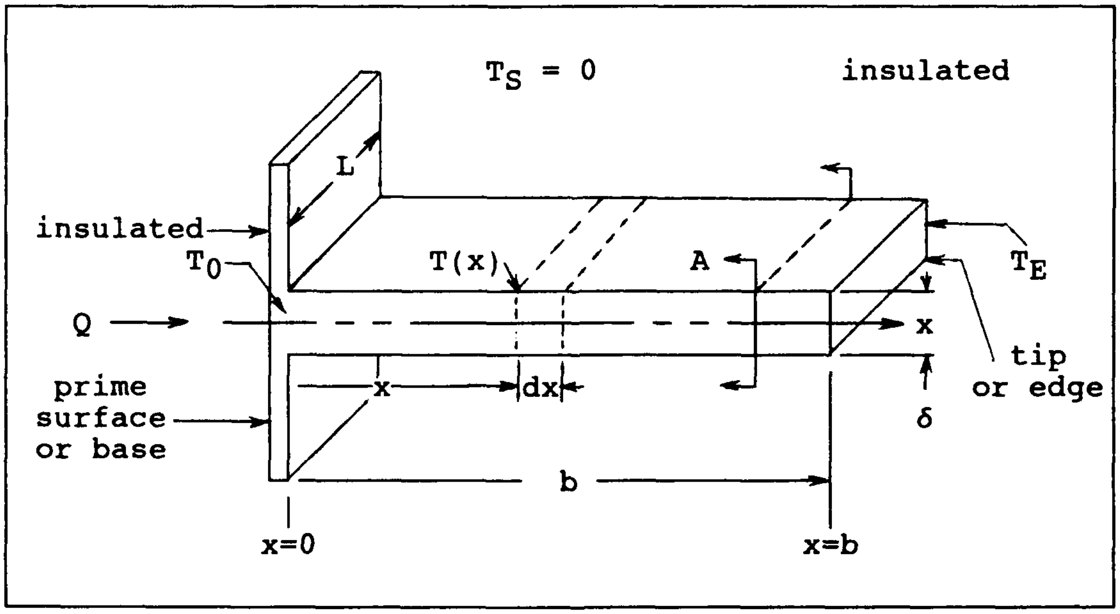

Figure 9 Heat Transfer in Rectangular Fin

Here:

Q = amount of heat transfer (W)

x = height coordinate along the fin with the positive orientation from base to tip (m)

dx = differential element (m)

$T(x)$ = fin temperature at x (^{o}K)

T_0 = temperature at fin base (^{o}K)

T_E = temperature at fin tip (^{o}K)

T_S = temperature of environment (for free space, $T_S = 0^{o}K$)

L = length of fin (m)

b = height of fin (m)

δ = thickness or width of fin (m)

A = cross-sectional area (m^2)

2. Assumptions

The following assumptions were made to simplify the mathematical development [Ref. 10: p. 196]:

1. The temperature surrounding the fin is uniform.
2. Steady-state.
3. No bond resistance exists between the fin and the prime surface.
4. There is no heat transfer through the fin tip.

6. The fin base temperature is uniform.
7. The fin width or thickness is small compared with the height and length of the fin.
8. The thermal conductivity of the fin material is isotropic and not a function of the temperature.

3. Analysis

For each differential element of the fin, the difference between heat entering and leaving by conduction is

$$dQ = k \frac{d}{dx} \left[A \frac{dT}{dx} \right] dx \qquad [4]$$

the heat dissipated by radiation is

$$dQ_R = 2 \sigma \epsilon L T^4 dx \qquad [5]$$

and the heat incident from external sources is

$$dQ_E = E \alpha L dx \qquad [6]$$

where:

E = external heat input (W/m^2)

α = surface absorptivity (dimensionless)

The absorptivities of some typical spacecraft materials are listed in Table III.

TABLE III ABSORPTIVITY

Material (9.3μm,310 °K)	Absorptivity
Black paint	0.90
Graphite/epoxy	0.84
Solar cells	0.70
Tiodized titanium	0.60
Aluminized kapton	0.35
Gold	0.25
White paint	0.20
Anodized aluminum	0.20
Aluminum	0.12
Optical solar reflector	0.08

Application of the general energy balance [1] to the differential element in steady-state results in

$$dQ = dQ_R - dQ_E \qquad [7]$$

Substituting equations [4] through [6] into equation [7] yields

$$k \frac{d}{dx}\left[A \frac{dT}{dx}\right] dx = 2\,\sigma\,\epsilon\,L\,T^4\,dx - E\,\alpha\,L\,dx \qquad [8]$$

With $A = \delta\ L$, $K_1 = 2\ \sigma\ \epsilon$ and $K_2 = E\ \alpha$ in equation [8], simplification gives

$$\frac{d^2T}{dx^2} - \frac{K_1}{k\ \delta}\ T^4 + \frac{K_2}{k\ \delta} = 0 \qquad [9]$$

Note that K_1 depends on the thermal property (emissivity) whereas K_2 depends on both a thermal property (absorptivity) and an external factor (external heat source). As previously discussed, external heat includes not only radiation incident from outside heat sources, but also radiation from the fin that is intercepted by nearby structures. In this sense, K_2 can be considered an environmental parameter. For the free-space case, $K_2 = 0$.

From Figure 9 and the assumption (no heat transfer through the tip of the fin), the boundary conditions are seen to be

$$T(0) = T_0$$
$$\left.\frac{dT}{dx}\right|_{x=b} = 0 \qquad [10]$$

C. TRAPEZOIDAL FIN

1. Geometry

The geometry, terminology and coordinate system for the longitudinal fin of trapezoidal profile are shown in Figure 10.

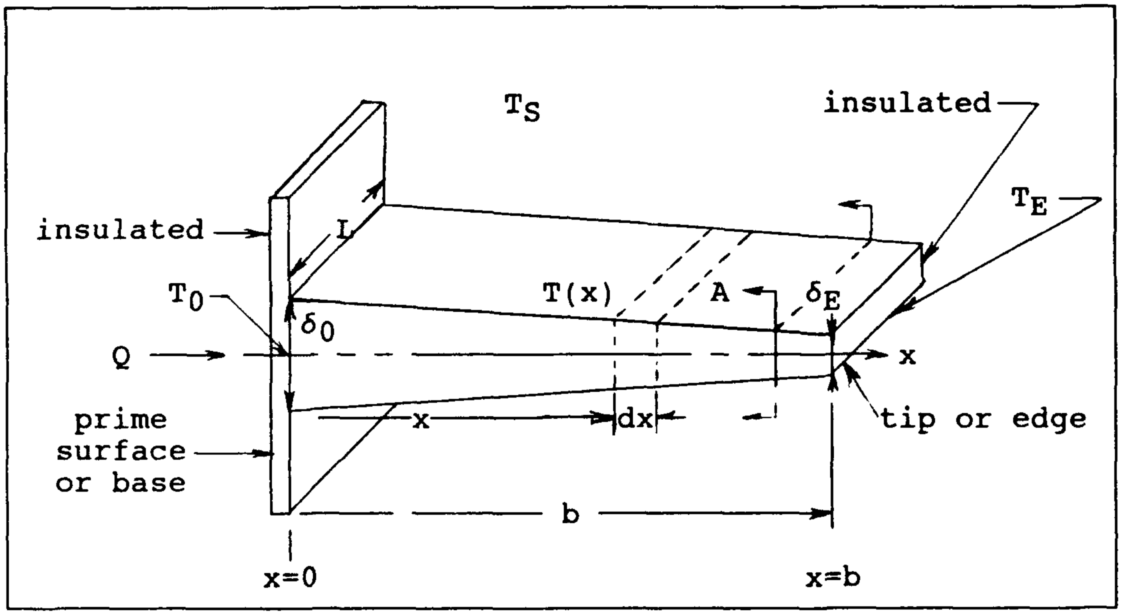

Figure 10 Heat Transfer in Trapezoidal Fin

Here:

Q = amount of heat transfer (W)

x = height coordinate along the fin with the positive orientation from base to tip (m)

dx = differential element (m)

T(x) = fin temperature at x (^{o}K)

T_0 = temperature at fin base (^{o}K)

T_E = temperature at fin tip (^{o}K)

T_S = temperature of environment (for free space, $T_S = 0^{o}K$)

L = length of fin (m)

b = height of fin (m)

$\delta(x)$ = thickness or width of fin at x (m)

δ_0 = thickness or width of fin at the base (m)

δ_E = thickness or width of fin at the tip (m)

2. Assumptions

The assumptions are the same as those for the longitudinal fin of rectangular profile.

3. Analysis

The analysis here is similar to that for the longitudinal fin of rectangular profile. It differs in that the fin thickness changes as a function of x:

$$\delta(x) = \delta_E + \frac{(\delta_0 - \delta_E)}{b} x \qquad [11]$$

For each differential element of the fin, the difference between the heat entering and leaving by conduction is

$$dQ = k\ L \frac{d}{dx} \left[\delta \frac{dT}{dx} \right] dx \qquad [12]$$

the heat leaving by radiation is

$$dQ_R = 2\ \sigma\ \epsilon\ L\ T^4\ dx \qquad [13]$$

and the heat incident from external sources is

$$dQ_E = E \alpha L \, dx \qquad [14]$$

Application of the general energy balance [1] to the differential element in steady-state establishes

$$dQ = dQ_R - dQ_E \qquad [15]$$

Substitution of equations [12] - [14] into equation [15] produces

$$k L \frac{d}{dx} \left[\delta \frac{dT}{dx} \right] dx = 2 \sigma \epsilon L T^4 dx - E \alpha L dx \qquad [16]$$

Then after substitution of $K_1 = 2 \sigma \epsilon$ and $K_2 = E \alpha$ into equation [16], simplification results in

$$k \frac{d}{dx} \left[\delta \frac{dT}{dx} \right] = K_1 T^4 - K_2 \qquad [17]$$

Differentiation of the left hand term gives

$$k \left[\delta \frac{d^2T}{dx^2} + \frac{d\delta}{dx} \frac{dT}{dx} \right] = K_1 T^4 - K_2 \qquad [18]$$

and then further simplification provides

$$\frac{d^2T}{dx^2} + \frac{1}{\delta} \frac{d\delta}{dx} \frac{dT}{dx} - \frac{K_1}{k \delta} T^4 + \frac{K_2}{k \delta} = 0 \qquad [19]$$

From Figure 10 and the assumptions, the boundary conditions are seen to be

$$T(0) = T_0$$
$$\left.\frac{dT}{dx}\right|_{x=b} = 0 \qquad [20]$$

D. **TRIANGULAR FIN**

The tip thickness for the longitudinal fin of trapezoidal profile is seen to be:

$$\delta(x) = \delta_E + \frac{(\delta_0 - \delta_E)}{b} x = \delta_E + \frac{(\delta_0 - \delta_E)}{b} b = \delta_E \qquad [21]$$

The longitudinal fin of triangular profile is the limiting case of the trapezoidal profile. The triangular fin has a theoretically "razor sharp" tip with zero thickness ($\delta_E = 0$). This results in a singularity at $\delta_E = 0$ in Equation 19. There are several methods available for handling such a singularity [Ref. 11: p. 48]:

1. Integration by parts.
2. Removal of the singularity by addition/subtraction.
3. Integration through a uniform approach to the limit.

However, because machining processes make it impossible to achieve a fin tip with zero thickness, the elimination of the singularity by one of the foregoing methods will not be

attempted. The approximation

$$\delta_E = 0.01 \times \delta_0 \qquad [22]$$

is more realistic from the standpoint of fin construction and also serves to eliminate the singularity at $x = 0$.

III. SOLUTION

A. PRELIMINARY

1. Review

The differential equations under consideration with their boundary conditions are:

For the rectangular profile:

$$\frac{d^2T}{dx^2} - \frac{K_1}{k\ \delta} T^4 + \frac{K_2}{k\ \delta} = 0 \qquad [23]$$

$$T(0) = T_0 \ ; \ \left.\frac{dT}{dx}\right|_{x=b} = 0 \qquad [24]$$

For the trapezoidal or triangular profiles:

$$\frac{d^2T}{dx^2} + \frac{1}{\delta}\frac{d\delta}{dx}\frac{dT}{dx} - \frac{K_1}{k\ \delta} T^4 \ \frac{K_2}{k\ \delta} = 0 \qquad [25]$$

$$T(0) = T_0 \ ; \ \left.\frac{dT}{dx}\right|_{x=b} = 0 \qquad [26]$$

Both Equations [23] and [25] are ordinary, second order, nonlinear differential equations. Equation [25], however, possesses variable coefficients. The presence of the T^4 term leads to the nonlinearity. The requirement to satisfy conditions at two different boundaries classifies

Equations [23] and [25] as boundary value problems. For a unique solution, a second order differential equation requires two boundary conditions. These types of problems are not easily solved by normal analytical methods but can be subjected to numerical procedures for solution.

2. Second Order Differential Equation

One procedure for solving a second order ordinary differential equation is to reduce it to a system of two first order differential equations (canonical form) by an appropriate substitution and then apply the techniques available for solving first order systems [Ref. 13: p. 379]. By letting Y = dT/dx in Equations [23] and [25], the second order ordinary differential equations are reduced to two first order systems :

$$\frac{dT}{dx} = Y$$
$$\frac{dY}{dx} = \frac{K_1}{k\,\delta} T^4 - \frac{K_2}{k\,\delta} \qquad [27]$$

with boundary conditions:

$$T(0) = T_0$$
$$Y(b) = 0 \qquad [28]$$

and

$$\frac{dT}{dx} = Y$$

$$\frac{dY}{dx} = -\frac{1}{\delta}\frac{d\delta}{dx}\frac{dT}{dx} + \frac{K_1}{k\,\delta}T^4 - \frac{K_2}{k\,\delta} \qquad [29]$$

with boundary conditions:

$$T(0) = T_0$$
$$Y(0) = 0 \qquad [30]$$

3. Boundary Value Problem

An <u>initial value problem</u> is a problem where known function and derivative values are given at the same point or boundary location. A <u>boundary value problem</u> is a problem where known function and derivative values are given at different boundary points or locations. There are numerous methods that can be used to solve initial value problems [Ref. 11: pp. 50 - 134]. These include:

1. Taylor Series Method
2. Euler's Method
3. Runge-Kutta Method
4. Multistep Method
5. Predictor-Corrector Method
6. Adams-Moulton Method

However, for boundary value problems there are basically two numerical methods [Ref. 11: p. 105]:

1. Finite Difference Method
2. Shooting Method

The finite difference method uses a set of difference equations to approximate the solution to the differential equation. The advantages to the finite difference method are its speed and economy of memory (solving a tridiagonal system of equations). However, this method runs into some difficulty when the differential equation is nonlinear because it results in a system of nonlinear finite difference equations that must be solved by an iterative technique. Thus the advantages of speed and economy of memory are largely eliminated.

The shooting method guesses the slope of the function at one end and then uses a standard integration scheme to solve the initial value problem to match the boundary condition at the other end (Figure 11) [Ref. 13: p. 412]. This procedure is repeated until the assumed solution meets the specified or known conditions at the boundaries. The advantage of the shooting method is its simplicity and its use of well established methods to solve initial value problems. Its disadvantage is that it is computationally intensive (dependent on the integration method used) and may take many guesses before meeting tolerance.

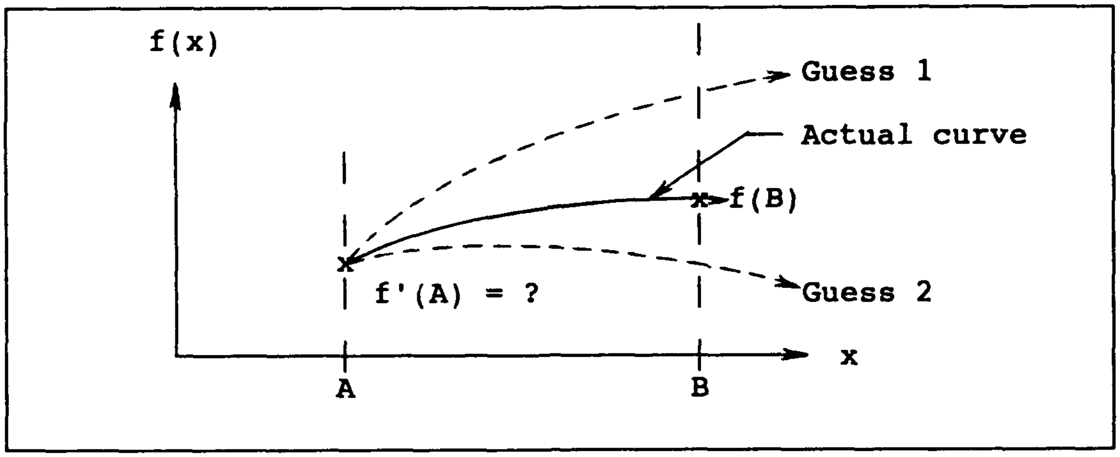

Figure 11 Shooting Method

B. THE ANALYSIS PROBLEM

For the analysis problem, the inputs are the fin parameters while the output is the amount of heat dissipated, Q. The boundary conditions are shown in Table IV.

TABLE IV ANALYSIS PROBLEM BOUNDARY CONDITIONS

Left	Right
$T(0) = T_0$	$T(b) = ? = T_E$
$\left.\frac{dT}{dx}\right\|_{x=0} = ?$	$\left.\frac{dT}{dx}\right\|_{x=b} = 0$

where: x = 0 at the base and x = b at the tip

Because the fin is in steady-state (there is no heat storage), the heat dissipated must equal the heat entering the base:

$$Q = -\,k\,A\left.\frac{dT}{dx}\right|_{x=0} \qquad [31]$$

Thus, if dT/dx is known at the base (x = 0), Q can be calculated. The procedure is to guess dT/dx at the base (x = 0) and use the shooting method to match the other known boundary condition dT/dx = 0 at the edge (x = b). This correct guess for dT/dx can be then substituted into Equation [31] to yield the heat dissipated, Q.

C. THE SYNTHESIS PROBLEM

For the synthesis problem, the input is the amount of heat to be dissipated while the output is the fin height required for its dissipation. Because the fin is in steady-state (there is no heat storage), the heat dissipated must equal the heat entering the base:

$$Q = - k A \left.\frac{dT}{dx}\right|_{x=0} \qquad [32]$$

However, unlike the analysis case, Q is known so dT/dx at x = 0 can be evaluated as

$$\left.\frac{dT}{dx}\right|_{x=0} = - \frac{k A}{Q} \qquad [33]$$

where x = 0 at the base and x = b at the edge.

Thus, in this case, the boundary conditions (see Figures 9 and 10) become

TABLE V SYNTHESIS PROBLEM BOUNDARY CONDITIONS

Left	Right
$T(0) = T_0$	$T(b) = ? = T_e$
$\left.\dfrac{dT}{dx}\right\vert_{x=0} = -\dfrac{k\,A}{Q}$	$\left.\dfrac{dT}{dx}\right\vert_{x=b} = 0$

Unfortunately, b, the height of the fin, which also provides the upper limit for integration, is unknown. The procedure here is to shoot from the known boundary condition $dT/dx = -kA/Q$ at $x = 0$ over the guessed interval (fin height, b) and match the other known boundary condition $dT/dx = 0$ at $x = b$.

D. ALGORITHM

1. Preliminary

For the analysis problem, the procedure can be viewed as the problem of finding the root of the function $w = F(z)$ where the independent variable, z, is the guess of the initial slope (dT/dx at $x = 0$) and the dependent variable, $w = F(z)$, is the slope, dT/dx at $x = b$.

A root finding method can then be applied to find the initial slope that will match the other boundary condition. Figure 12 illustrates this relationship.

In similar fashion, for the synthesis problem, the procedure can be viewed as the problem of finding the root of the function $w = F(z)$ where the independent variable, z,

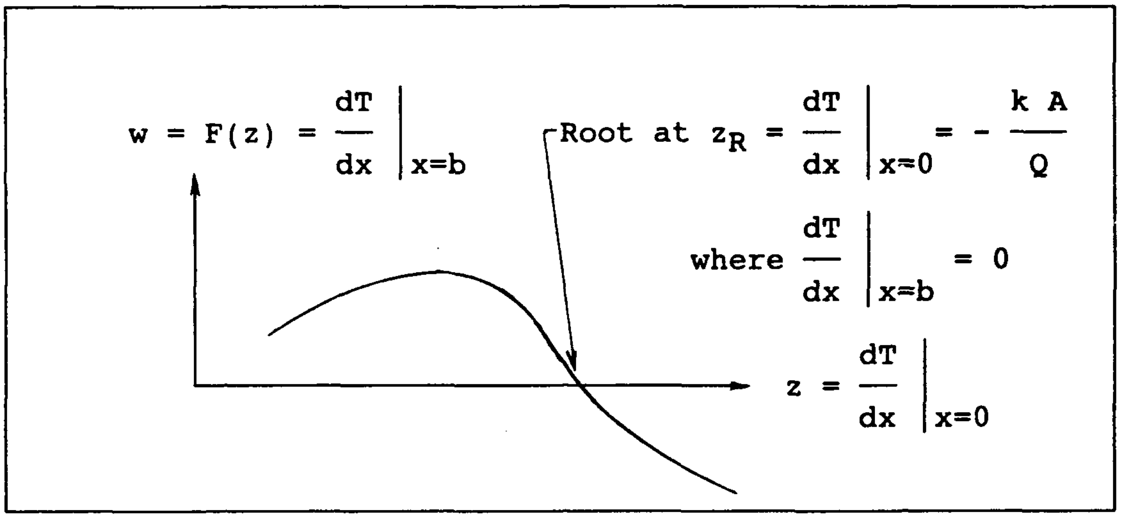

Figure 12 Analysis Problem as a Root Finding Problem

is the guess at the height of the fin, b, and the dependent variable, F(z) is once again the final slope, dT/dx at x = b, at the other end. Then a root finding method can again be applied to find the height that will give the correct boundary condition at the fin tip. Figure 13 illustrates this relationship.

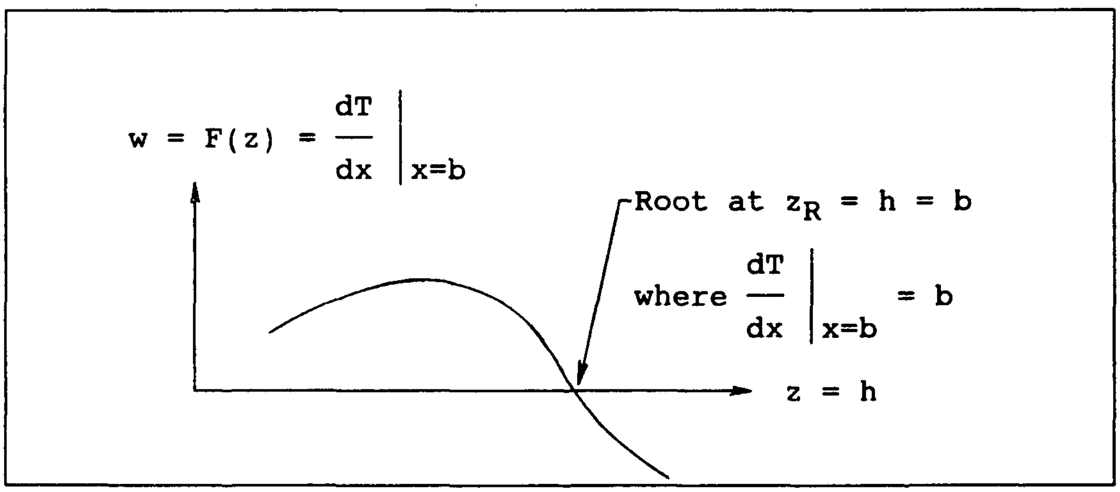

Figure 13 Synthesis Problem as a Root Finding Problem

2. Modified Linear Interpolation

There are many root finding methods available for finding roots of nonlinear equations [Ref. 14: p. 27]:

1. Bisection Method
2. Linear Interpolation Method
3. Modified Linear Interpolation Method
4. Secant Method
5. Newton's Method
6. Fixed Point Iteration Method
7. Muller's Method

The root finding method used here is the modified linear interpolation method (Figure 14). It offers a good combination of accuracy and speed in finding the root of an equation.

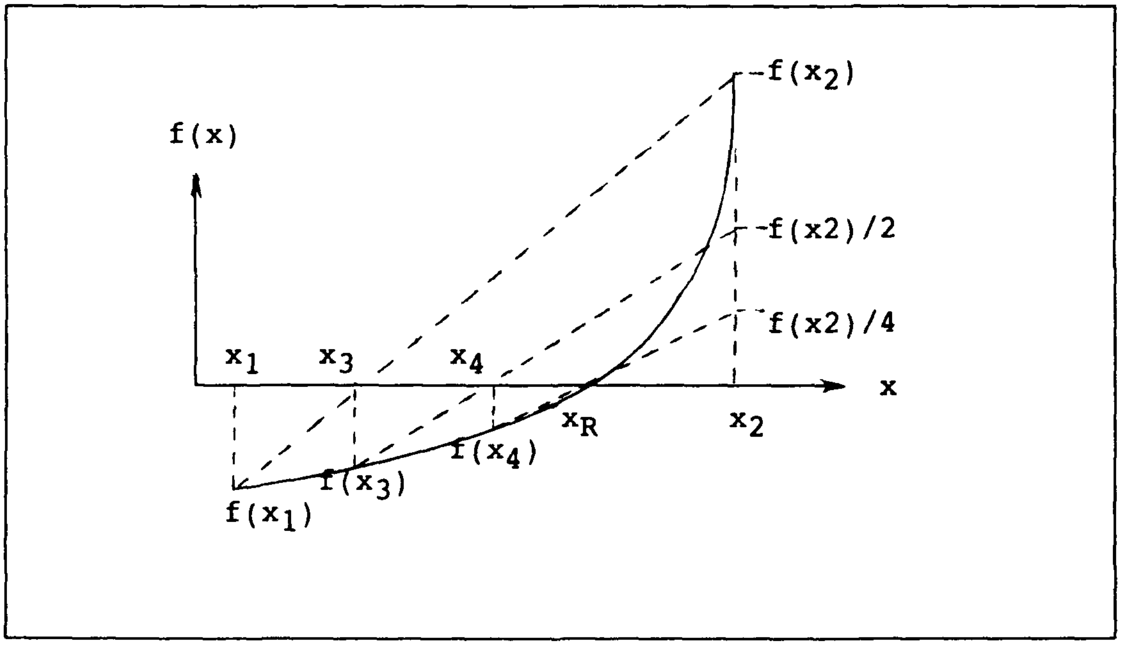

Figure 14 Modified Linear Interpolation

The algorithm is described in pseudocode as shown in Figure 15 [Ref. 13: p. 11].

```
Let x1 and x2 bracket the root; ie., f(x1) and
    f(x2) are opposite in sign

Set SAVE = f(x1); set F1 = f(x2); set F2 = f(x2)

Repeat
```

$$\text{Set } x_3 = x_2 - F2\,\frac{x_2 - x_1}{F2 - F1}$$

```
    If f(x3) of opposite sign to F1

       Set x2 = x3; set F2 = f(x3)

       If f(x3) of same sign as SAVE

          Set F1 = F1/2

        Endif

    Else

       Set x1 = x3; set F1 = f(x3)

       If f(x3) of same sign as SAVE

         Set F2 = F2/2

         Endif

    Endif

    Set SAVE = f(x3)

Until |x1 - x2| < XTOL or |f(x3)| < FTOL
```

Figure 15 Modified Linear Interpolation Pseudocode

where:

x_1 = x value at left end point

x_2 = x value at right end point

x_3 = x value at intermediate point

$f(x_1)$ = function value at left end point

$f(x_2)$ = function value at right end point

$f(x_3)$ = function value at intermediate point

F1 = saved value of $f(x_1)$

F2 = saved value of $f(x_2)$

SAVE = function used for next sign comparison

XTOL = tolerance in independent variable, x

FTOL = tolerance in dependent variable, f(x)

Note that halving the functional evaluations (F1/2 or F2/2) alleviates the difficulty encountered by regular linear interpolation on one-sided approaches to the root as shown in Figure 16.

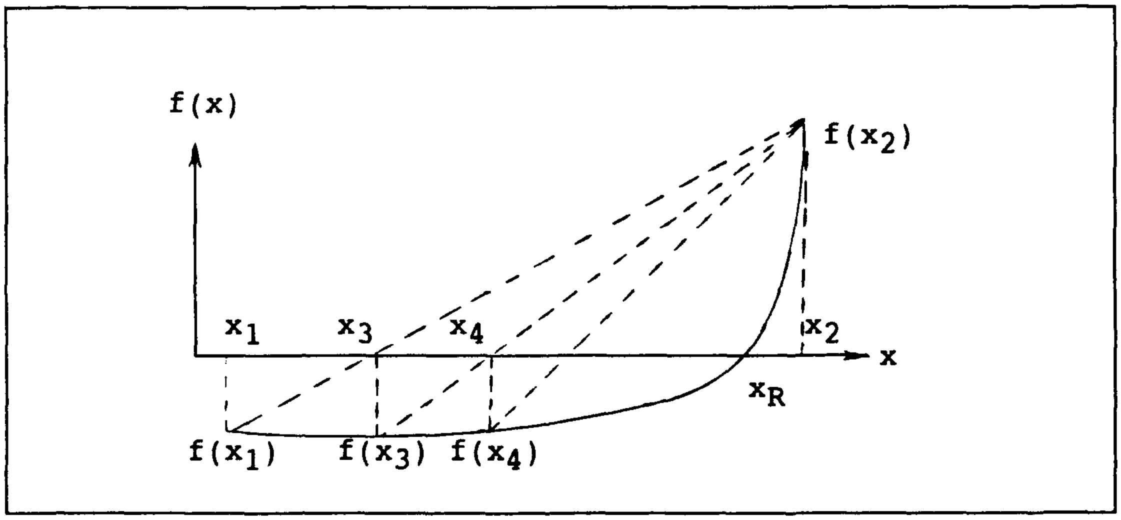

Figure 16 One-sided Approach to Root

One of the major difficulties in executing a root solver is to find an interval that brackets the root (bisection, linear interpolation, modified linear

interpolation, or secant) or that is reasonably close to the root (Newton, fixed point iteration, or Muller). The method used here is a simple linear search from the origin as shown in Figure 17 [Ref. 15: p. 42].

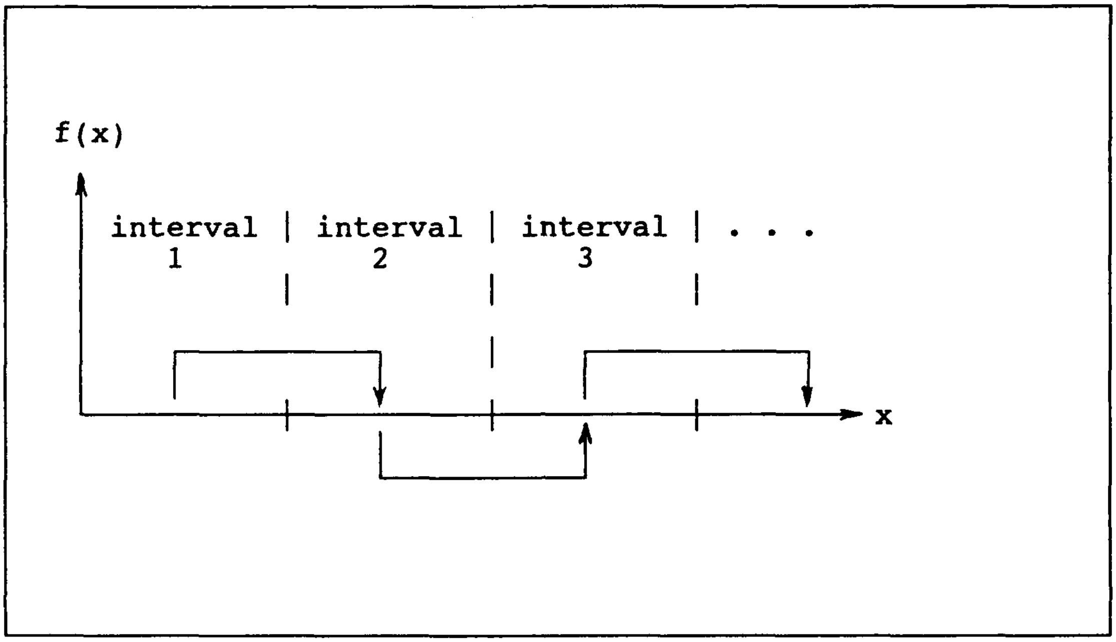

Figure 17 Search for Interval to Bracket Root

For the synthesis problem, the independent variable is $z = -dT/dx$ and so the algorithm searches successive intervals [0,-1], [-1,-2], ... for functional values that will be opposite in sign. Here, the independent variable must be negative to ensure that Q, the amount of heat, is by convention a positive quantity. The unit interval search is a compromise between too small an interval that would require too many searches and too large an interval which may miss the root (Figure 18).

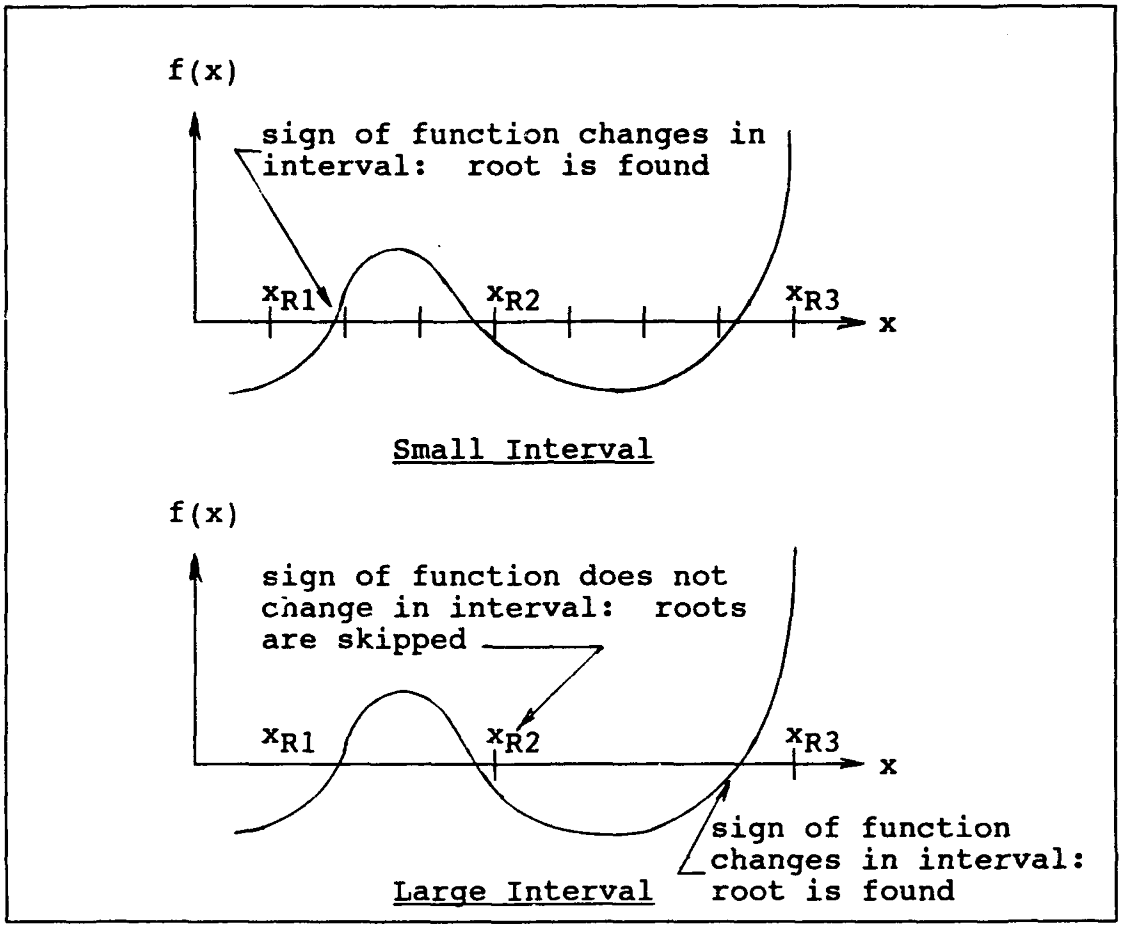

Figure 18 Root Search for Multiple Roots

3. Runge-Kutta-Fehlberg

The shooting method requires an integrating scheme to "march" or "shoot" from one boundary condition to the other. As mentioned previously, there are many techniques of numerical integration:

1. Taylor Series Method
2. Euler's Method
3. Modified Euler's Method
4. Runge-Kutta Method

5. Milne Method

6. Adams-Moulton Method

Runge-Kutta-Fehlberg offers the best combination of accuracy, stability and adaptability as indicated in Table VI.

TABLE VI COMPARISON OF METHODS FOR SOLVING DIFFERENTIAL EQUATIONS

Method	Global Error	Function Evaluations Per Step	Stability	Ease of Step Size Adjustment
Modified Euler	$O(h^2)$	2	G	G
4TH-order R-K	$O(h^5)$	4	G	G
R-K-Fehlberg	$O(h^6)$	6	G	G
Milne	$O(h^5)$	2	P	P
Adams-Moulton	$O(h^5)$	2	G	P

where:

R-K = Runge-Kutta

h = step size

$O(h^2)$ = error on the order of h^2

G = good

P = poor

The Runge-Kutta-Fehlberg method uses a 4TH-order Runge-Kutta method to produce one estimate (y'_{N+1}) and uses a 5TH-order Runge-Kutta method to produce another estimate (y_{N+1}). It then computes the error $E = y_{N+1} - y'_{N+1}$ and adjusts the step, h, accordingly while using the y_{N+1} as the

next estimate. Such a scheme is called <u>adaptive</u> or <u>smart</u> because it capable of estimating the error at each step and making a corresponding adjustment in step size to remain within the tolerance. Because the k's are the same for both estimates, only six functional evaluations are needed. The power of the method is its high accuracy, $O(h_5)$, and adaptability. A disadvantage is that it requires more functional evaluations than other methods. The coefficients are difficult to derive algebraically but can be thought of as weights applied to previous estimates that are added to produce a future estimate. These values are listed in Table VII.

TABLE VII RUNGE-KUTTA-FEHLBERG COEFFICIENTS

$$k_1 = h\ f(x_N, y_N)$$

$$k_2 = h\ f(x_N + h/4, y_N + k_1/4)$$

$$k_3 = h\ f(x_N + 3h/8, y_N + 3k_1/32 + 9k_2/32)$$

$$k_4 = h\ f(x_N + 12h/13, y_N + 1932k_1/2197 - 7200k_2/2197 + 7296k_3/2197)$$

$$k_5 = h\ f(x_N + h, y_N + 439k_1/216 - 8k_2 + 3680k_3/513 - 845k_4/4104)$$

$$k_C = h\ f(x_N + h/2, y_N - 8k_1/27 + 2k_2 - 3544k_3/2565 + 1859k_4/4104 - 11k_5/40)$$

$$y'_{N+1} = y_N + (25k_1/216 + 1408k_3/2565 + 2197k_4/4104 - k_5/5) \text{ with global error } O(h_4)$$

$$y_{N+1} = y_N + (16k_1/135 + 6656k_3/12825 + 28561k_4/54430 - 9k_5/50 + 2k_6/55) \text{ with global error } O(h_5)$$

$$E = y_{N+1} - y'_{N+1} = k_1/360 - 128k_3/4275 - 2197k_4/75240 + k_5/50 + 2k_6/55$$

where:

h = step size

x_N = current independent variable, x

y_N = current dependent variable, y

f = derivative function = $f(x,y)$

$k_1 - k_6$ = weighted values of f along interval

y'_{N+1} = predicted value of y using 4TH order Runge-Kutta integration scheme

y_{N+1} = predicted value of y using 5TH order Runge-Kutta integration scheme

$O(h_4)$ = error on the order of h_4

$O(h_5)$ = error on the order of h_5

E = error between 4TH and 5Th order predicted values

IV. COMPUTER PROGRAM

A. PRELIMINARY

1. Structure Design

The computer program design follows the concept of structured design. Meilir Page-Jones in his handbook on structure design describes it as "the development of a blueprint of a computer system solution to a problem that has the same components and interrelationships among the components as the original problem" [Ref. 16: p. 3].

The structure design concept has seven major steps [Ref. 16: p. 20]:

1. Problem recognition
2. Feasibility Study
3. Analysis
4. Design
5. Implementation
6. Testing
7. Maintenance

The previous sections dealt with steps one and three. Steps two, five and seven refer to more intensive software design projects and are not applicable here. The remaining sections will concentrate on steps four and six.

2. Problem Review

Recall that the original problem consisted of the following elements:

1. Environments: free space, non-free space
2. Fin profiles: rectangular, trapezoid, triangular
3. Problem types: analysis, synthesis

Non-free space includes external heat sources. The analysis problem determines the amount of heat dissipated by a fin of given dimensions. The synthesis problem determines the size of the fin to dissipate a given quantity of heat.

3. Problem Reduction

Because the free space case is just a special case of the non-free space, its solution will be included in the more general case ($K_2 = E\ \alpha = 0$ in Equations [23] and [25]).

With this simplification, the problem can be subdivided into six "sub-problems" based on geometry, differential equation and input/outputs as listed in TABLE VIII. Although the triangular fin is a special case of the trapezoidal fin (with tip thickness equal to 1/100th of base thickness), it is still handled separately to take advantage of program modularity which is discussed later.

4. Program Inputs/Outputs

In view of the input/output format of computer subroutines, the sub-problems can also be analyzed from the standpoint of inputs/outputs as listed in TABLE IX.

TABLE VIII SUB-PROBLEM TYPES

Nr	Profile	Problem
1	Rectangle	Analysis
2	Rectangle	Synthesis
3	Trapezoid	Analysis
4	Trapezoid	Synthesis
5	Triangle	Analysis
6	Triangle	Synthesis

TABLE IX INPUT/OUTPUT RELATIONSHIPS

Nr	Shape	Type	Input	Output
1	Rectangle	Anal	$T_0, L, \delta, k, \alpha, \epsilon, E, b$	$Q, T_E, \cap$
2	Rectangle	Syn	$T_0, L, \delta, k, \alpha, \epsilon, E, Q$	$b, T_E, \cap$
3	Trapezoid	Anal	$T_0, L, \delta_0, \delta_E, k, \alpha, \epsilon, E, b$	$Q, T_E, \cap$
4	Trapezoid	Syn	$T_0, L, \delta_0, \delta_E, k, \alpha, \epsilon, E, Q$	$b, T_E, \cap$
5	Triangle	Anal	$T_0, L, \delta_0, k, \alpha, \epsilon, E, b$	$Q, T_E, \cap$
6	Triangle	Syn	$T_0, L, \delta_0, k, \alpha, \epsilon, E, Q$	$b, T_E, \cap$

where:

Anal - analysis

Syn - synthesis

T_0 - temperature at fin base

L - fin length

δ - fin width or thickness (rectangle)

δ_0 - fin width or thickness at base (trapezoid or triangle)

δ_E - fin width or thickness at tip (trapezoid)

k - conductivity of fin

α - absorptivity of fin

ϵ - emissivity of fin

Q - real heat dissipated by the fin

b - fin height

E - external heat flux

T_E - temperature at fin tip

η - fin efficiency

B. FLOWCHART

The flow chart logic is shown in Figure 19. The first step is to perform housekeeping chores such as data initialization and file access. Next the units, fin profile and problem type are selected which in turn specifies the fin parameters to be input. The program then branches to the subroutine that solves one of the above six sub-problems. Each subprogram finds the root of equation involving one of the desired outputs. This requires solving the associated second order, nonlinear differential equation describing the heat transfer under the specified boundary conditions. The output is then summarized and presented. The same problem can be repeated with different outputs or a new problem can be entered or the program can be terminated. The abbreviations used in Figure 19 are:

RECT = rectangle

TRAP = trapezoid

TRI = triangle

ANAL = analysis

SYN = synthesis

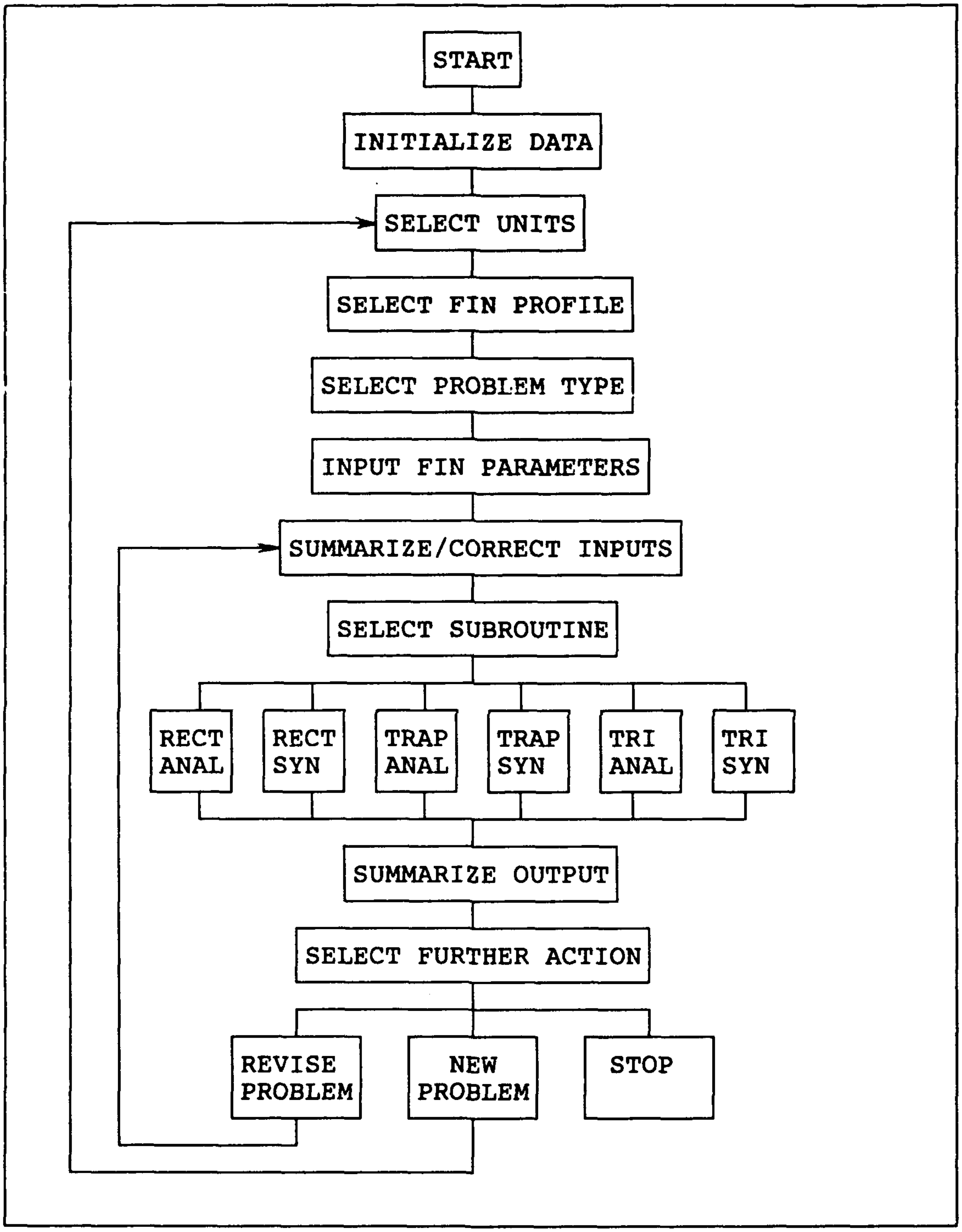

Figure 19 Flowchart

C. MODULES

The basic principals of the design step are:

1. Partition the system into modules.
2. Organize the modules into a hierarchal structure.

The principal tool used to do this is the structure chart. The structure chart "illustrates the partitioning of a system into modules - showing their hierarchy, organization, and communication" [Ref. 12: p. 10].

Applying this concept to Tables VIII and IX, the structure chart takes the form as shown in Figure 20. The chart shows the basic modules and their relationships. The program for fin analysis is divided into multi-level modules whose attributes are given in Table X. The level refers to the program hierarchy as shown in Table XI.

D. SPECIFICATIONS

The program is written in FORTRAN-77 for IBM-compatible computers. It uses only very basic FORTRAN statements common to most FORTRAN compilers to enhance compatibility with a wide variety of computer environments. The program has been compiled and executed in the PC environment with Microsoft's Fortran Compiler. It has also been run on the mainframe (IBM 360/370) with the WF77 and FORTVS compilers.

The program is written in double precision to minimize round-off and truncation errors, individually and globally, that arise in the computationally intensive routines (root

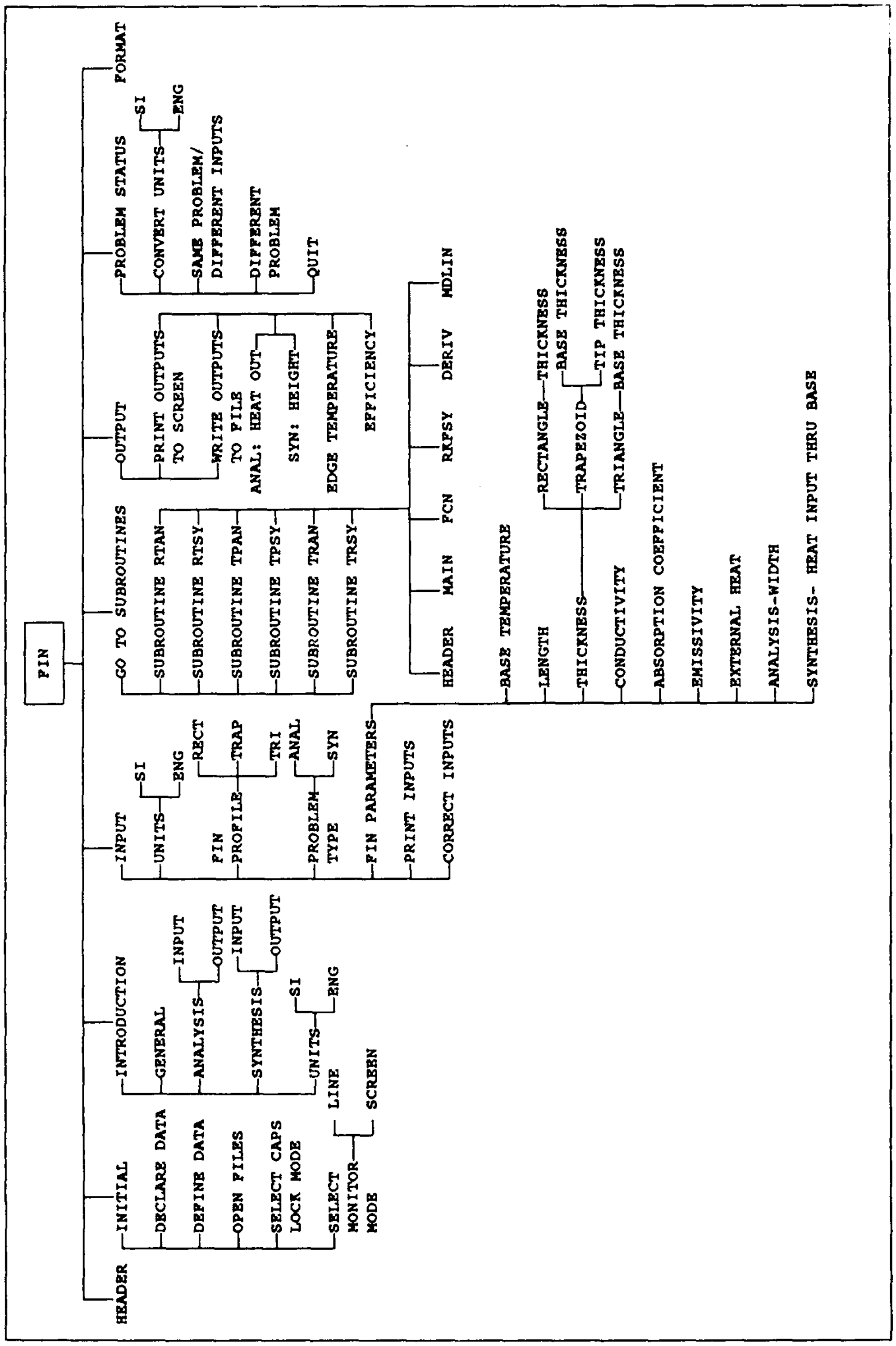

Figure 20 Computer Program Modules

TABLE X MODULES

Module	Level	Action
HEADER	1	Provides program documentation
INITIAL	1	Declares variables Sets initial values Opens file for data Selects monitor modes
INTRO	1	Presents info menus: - overview - environments - profiles - problem types - units
INPUT	1	Processes input for: - units - profile - problem type - fin parameters Summarizes data input Corrects data input
GOTO	1	Based on inputs chooses one of six subroutines Performs conversions as required
RTAN	2	Solves rectangle, analysis problem
RTSY	2	Solves rectangle, synthesis problem
TPAN	2	Solves trapezoid, analysis problem
TPSY	2	Solves trapezoid, synthesis problem
TRAN	2	Solves triangle, analysis problem
TRSY	2	Solves triangle, synthesis problem
OUTPUT	1	Prints outputs to screen Writes outputs to FIN.DAT file
PGMSTAT	1	Performs conversion as requested Sets up same problem with different inputs specified by user Sets up new problem Closes and saves data file Exits program
FORMAT	1	Input/output format
SI	2	Converts to SI units
ENG	2	Converts to English units
FCN	4	Computes function values
RKFSY	5	Solves system of differential equations by Runge-Kutta-Fehlberg
DERIV	6	Defines derivatives
MDLIN	3	Solves for root of function by modified linear interpolation

TABLE XI PROGRAM HIERARCHY

Level	Hierarchy
1	Main program level
2	Subprogram called from level 1
3	Subprogram called from level 2
4	Subprogram called from level 3
5	Subprogram called from level 4
6	Subprogram called from level 5

solver and integrator). Most inputs will have at best 4 - 6 significant figures due to physical measurement limitations, so the accuracy of the results can not be expected to exceed that.

The program is completely independent and invokes no external library routines. All numerical subroutines for root finding and integration have been incorporated into the program.

The simplicity mentioned above has been achieved at the cost of program length - nearly 4500 lines of code. Additional length also resulted from the duplication of some sections to maintain program modularity. For development and subsequent improvement, it is important that individual modules have the capability of being removed, modified and reinserted with minimum interference. To this end, all main program elements and sub-elements can be easily removed and independently operated with simple "drivers".

Program specifications are summarized in Table XII.

TABLE XII PROGRAM SPECIFICATIONS

Parameter	Specification
Language	FORTRAN 77
Lines of code	4425
Memory: Fortran code Object code Execution code Total:	140020 102455 107370 349845
Compiler	Microsoft FORTRAN V5.0
Compile time	3.5 minutes
Computer	IBM clone (386/387/16.0Mhz)
Execution time	Variable (see examples)
Precision	Double
External programs/ libraries required	None

E. MENUS

The program is executed in the interactive mode. The user interfaces with the program through a series of menus. The two interfacing modes are:

1. Line edit
2. Screen edit

In the line edit mode, the user interacts with the program's commands as they are "scrolled" to the screen. This enables the last few commands to always be on the screen. In the screen edit mode, there is one command per screen and the screen is cleared before the next command, thereby minimizing clutter on the screen. The screen edit

mode can only by used on a typical 24 line PC monitor. It will not "synchronize' with other monitors or with the video terminals associated with the mainframe. The line edit mode should be used in these cases.

The menus are straightforward and consist of simple questions or commands to enter inputs. Table XIII is a listing of the menus.

TABLE XIII MENUS

NR	Query/Input
1	"CAPS LOCK" selection
2	Viewing mode selection
3	Introduction
4	Units selection
5	Fin profile selection
6	Problem type selection
7	Base temperature input
8	Length input
9	Thickness input
10	Conductivity input
11	Absorptivity input
12	Emissivity input
13	External heat input
14	Heat entering base (synthesis) input
15	Height (analysis) input
16	Incorrect entry input
17	Output conversion
18	Problem selecticn: same problem with changes
19	Problem selection: new problem
20	Quit

F. ERROR ROUTINES

Several error routines have been built into the program. These can be roughly divided into three major categories:

1. Input error detection
2. Output error detection
3. Program error detection

The first category concerns errors made by the operator during the course of menu interrogation. Although no program can be completely "idiot proof", an attempt has been made to minimize input errors by restricting the input range to realistic values as shown in Table XIV. Any input outside this range will result in an "INCORRECT ENTRY" flag along with a repetition of the menu command.

TABLE XIV INPUT PARAMETER RANGE

Parameter	Range
Y or N	Capital "Y" or capital "N"
1,2,...	Numbered response "1","2",...
Temp at fin base (T_0)	$\geq -273.15^{\circ}$K (-460.0°F)
Fin length (L)	> 0
Base thickness (δ or δ_0)	> 0
Tip thickness (δ_E)	> 0
Conductivity (k)	> 0
Absorptivity (α)	$0 \leq \alpha \leq 1$
Emissivity (ϵ)	$0 \leq \epsilon \leq 1$
External heat flux (E)	≥ 0
Heat input thru base (Q)	> 0
Fin height (b)	> 0

Similarly, any values outside the ranges for the outputs listed in Table XV will result in a flag to check the inputs for correctness.

Finally, there are internal error checks to minimize the possibility of a "hung" program. Here, the term "hung" program means that the algorithm is caught in an "infinite"

TABLE XV OUTPUT PARAMETER RANGE

Parameter	Range
Heat input thru base (Q)	> 0
Fin height (b)	> 0
Temp at fin tip (T_E)	$\geq -273.15^{O}K, -460.0^{O}F$
Efficiency ($\cap$)	$0 \leq \cap \leq 1$

loop; i.e., the problem oscillates or diverges. No numerical procedure can or should be made so "robust" that it converges regardless of the inputs. In this sense, the best defense is to ensure that the inputs are not only restricted to the ranges listed above, but that they also appear reasonable. Fins with thicknesses of a hundred meters or with base temperatures of ten thousand degrees are unreasonable.

The three major numerical algorithms (interval search, root solver and differential equation solver) have several methods of stopping. The interval search routine is limited to a fixed number of intervals (100) to be searched. If a root is not found within the first 100 intervals, the user is asked if he wants to continue the search for another 100 intervals. This procedure is repeated until an interval is found or the user's patience is exhausted. The user may also request that the function values be printed to the screen. This will give some idea of the function's behavior. Recall that for a function to have a root at x, f(x) must equal zero and hence any interval that brackets

the root must have function values at the endpoints that are opposite in sign. So by examining the function values, the user can get a "feel" for whether or not the function values are approaching zero with a resultant sign change. Note that this is not an absolute guarantee since a function can increase then decrease or vice versa.

Both the root and differential equation solvers terminate through one of the following mechanisms:

1. Successive estimated values are within a specified tolerance.

2. A fixed number of iterations is exceeded.

The termination criteria are listed in Table XVI. As is the case for the interval search algorithm, the user can view the function values to get a rough idea of convergence or divergence.

TABLE XVI ROOT SOLVER AND DIFFERENTIAL EQUATION SOLVER STOPPING CRITERIA

Routine	Parameter	Stopping criteria
Modified linear interpolation	X tolerance	0.0001
	F(X) tolerance	0.00001
	Iterations	50
Runge-Kutta-Fehlberg	Y tolerance	0.0001
	Iterations	100

V. EXAMPLES

A. PRELIMINARY

The following examples demonstrate the operation of the program. The results compared favorably with "manual" plots [Ref 7, pp. 242 - 272]. The input/output summaries are copies of ones that were actually written to the file FIN.DAT. Note that the synthesis problems are intentionally "inverses" of the analysis problems so as to further check results. Also the execution times on an IBM clone (386/387/16.0 Mhz) are included in parenthesis for comparison. The times are a function of the machine as well as the problem itself (geometry and fin parameters). The triangular fin/synthesis problem is particularly slow because of the small step used in integration.

B. RECTANGULAR FIN

1. Analysis (29 seconds)

A rectangular fin has a base temperature of 77 ^{O}C, a length of 4 meters, a thickness of 0.635 centimeters, a height of 50 centimeters, a conductivity of 152 Watts/meter, an absorptivity of zero, and an emissivity of 0.85. There are no external heat sources. What is the amount of heat dissipated by the fin? What is the temperature at the fin tip? What is the fin efficiency?

INPUT/OUTPUT SUMMARY:

```
*********************** INPUT ****************************

 (1) UNITS                    =    SI
 (2) FIN                      =    RECTANGLE
 (3) PROBLEM                  =    ANALYSIS
 (4) BASE TEMPERATURE         =    77.0000      C
 (5) FIN LENGTH               =    4.00000      M
 (6) BASE THICKNESS           =    .635000E-02 M
 (7) CONDUCTIVITY CONSTANT    =    152.000      W/M-C
 (8) ABSORPTION COEFFICIENT=       .000000
 (9) EMISSIVITY               =    .850000
(10) EXTERNAL HEAT FLUX       =    .000000      W
(11) FIN HEIGHT               =    .500000      M

*********************** OUTPUT ***************************

(12) HEAT INPUT THRU BASE     =     1510.83     W
(13) EDGE TEMPERATURE         =    -3.32502     C
(14) FIN EFFICIENCY           =     .521312
```

2. **Synthesis** (43 seconds)

A rectangular fin has a base temperature of 77 ^{O}C, a length of 4 meters, a thickness of 0.635 centimeters, a conductivity of 152 Watts/meter, an absorptivity of zero, and an emissivity of 0.85. There are no external heat sources. What is the height required for the fin to dissipate 1510.83 Watts of heat entering through the base (see the previous problem)? What is the temperature at the fin tip? What is the fin efficiency?

INPUT/OUTPUT SUMMARY:

```
*********************** INPUT ****************************

 (1) UNITS                    =    SI
 (2) FIN                      =    RECTANGLE
 (3) PROBLEM                  =    SYNTHESIS
 (4) BASE TEMPERATURE         =    77.0000      C
 (5) FIN LENGTH               =    4.00000      M
 (6) BASE THICKNESS           =    .635000E-02 M
```

```
 (7) CONDUCTIVITY CONSTANT =    152.000     W/M-C
 (8) ABSORPTION COEFFICIENT=    .000000
 (9) EMISSIVITY            =    .850000
(10) EXTERNAL HEAT FLUX    =    .000000     W
(11) HEAT INPUT THRU BASE  =    1510.83     W

********************** OUTPUT **************************

(12) FIN HEIGHT            =    .500015     M
(13) EDGE TEMPERATURE      =   -3.32527     C
(14) FIN EFFICIENCY        =    .521297
```

C. TRAPEZOIDAL FIN

1. Analysis (9 seconds)

A trapezoidal fin has a base temperature of 167 ^{O}C, a length of 1.143 meters, a base thickness of 0.9525 centimeters, an edge thickness of 0.47625 centimeters, a height of 15.24 centimeters, a conductivity of 33 Watts/meter, an absorptivity of zero, and an emissivity of 0.95. There are no external heat sources. What is the amount of heat dissipated by the fin? What is the temperature at the fin tip? What is the fin efficiency?

INPUT/OUTPUT SUMMARY:

```
********************** INPUT ***************************

 (1) UNITS                 =    SI
 (2) FIN                   =    TRAPEZOID
 (3) PROBLEM               =    ANALYSIS
 (4) BASE TEMPERATURE      =    167.000     C
 (5) FIN LENGTH            =    1.14300     M
 (6) BASE/EDGE THICKNESSES =    .952500E-02   .476250E-02 M
 (7) CONDUCTIVITY CONSTANT =    33.0000     W/M-C
 (8) ABSORPTION COEFFICIENT=    .000000
 (9) EMISSIVITY            =    .950000
(10) EXTERNAL HEAT FLUX    =    .000000     W
(11) FIN HEIGHT            =    .152400     M
```

```
********************** OUTPUT *************************

(12) HEAT INPUT THRU BASE  =    412.048     W
(13) EDGE TEMPERATURE      =    77.7946     C
(14) FIN EFFICIENCY        =    .584976
```

2. Synthesis (11 seconds)

A trapezoidal fin has a base temperature of 167 ^{O}C, a length of 1.143 meters, a base thickness of 0.9525 centimeters, an edge thickness of 0.47625 centimeters, a conductivity of 33 Watts/meter, an absorptivity of zero, and an emissivity of 0.95. There are no external heat sources. What is the height required for the fin to dissipate 412.048 Watts of heat entering through the base (see the previous problem)? What is the temperature at the fin tip? What is the fin efficiency?

```
INPUT/OUTPUT SUMMARY:

********************** INPUT **************************

 (1) UNITS                 =    SI
 (2) FIN                   =    TRAPEZOID
 (3) PROBLEM               =    SYNTHESIS
 (4) BASE TEMPERATURE      =    167.000     C
 (5) FIN LENGTH            =    1.14300     M
 (6) BASE/EDGE THICKNESSES =    .952500E-02    476250E-02 M
 (7) CONDUCTIVITY CONSTANT =    33.0000     W/M-C
 (8) ABSORPTION COEFFICIENT=    .000000
 (9) EMISSIVITY            =    .950000
(10) EXTERNAL HEAT FLUX    =    .000000     W
(11) HEAT INPUT THRU BASE  =    412.048     W

********************** OUTPUT *************************

(12) FIN HEIGHT            =    .152399     M
(13) EDGE TEMPERATURE      =    77.7945     C
(14) FIN EFFICIENCY        =    .584980
```

D. TRIANGULAR FIN

1. **Analysis** (14 seconds)

A triangular fin has a base temperature of 167 ^{O}C, a length of 1.143 meters, a base thickness of 0.9525 centimeters, a height of 16.5 centimeters, a conductivity of 33 Watts/meter, an absorptivity of zero, and an emissivity of 0.95. There are no external heat sources. What is the amount of heat dissipated by the fin? What is the temperature at the fin tip? What is the fin efficiency?

INPUT/OUTPUT SUMMARY:

```
********************** INPUT ***************************

 (1) UNITS                  =    SI
 (2) FIN                    =    TRIANGLE
 (3) PROBLEM                =    ANALYSIS
 (4) BASE TEMPERATURE       =    167.000      C
 (5) FIN LENGTH             =    1.14300      M
 (6) BASE THICKNESS         =    .952500E-02  M
 (7) CONDUCTIVITY CONSTANT  =    33.0000      W/M-C
 (8) ABSORPTION COEFFICIENT=     .000000
 (9) EMISSIVITY             =    .950000
(10) EXTERNAL HEAT FLUX     =    .000000      W
(11) FIN HEIGHT             =    .165000      M

********************** OUTPUT **************************

(12) HEAT INPUT THRU BASE   =    400.040      W
(13) EDGE TEMPERATURE       =    40.4865      C
(14) FIN EFFICIENCY         =    .524560
```

2. **Synthesis** (3.7 minutes)

A triangular fin has a base temperature of 167 ^{O}C, a length of 1.143 meters, a base thickness of 0.9525 centimeters, a conductivity of 33 Watts/meter, an absorptivity of zero, and an emissivity of 0.95. There are

no external heat sources. What is the height required for the fin to dissipate 400.040 Watts of heat entering through the base (see the previous problem)? What is the temperature at the fin tip? What is the fin efficiency?

INPUT/OUTPUT SUMMARY:

```
********************** INPUT ***************************

 (1) UNITS                    =    SI
 (2) FIN                      =    TRIANGLE
 (3) PROBLEM                  =    SYNTHESIS
 (4) BASE TEMPERATURE         =    167.000      C
 (5) FIN LENGTH               =    1.14300      M
 (6) BASE THICKNESS           =    .952500E-02  M
 (7) CONDUCTIVITY CONSTANT    =    33.0000      W/M-C
 (8) ABSORPTION COEFFICIENT=       .000000
 (9) EMISSIVITY               =    .950000
(10) EXTERNAL HEAT FLUX       =    .000000      W
(11) HEAT INPUT THRU BASE     =    400.040      W

********************** OUTPUT *************************

(12) FIN HEIGHT               =    .165480      M
(13) EDGE TEMPERATURE         =    42.2041      C
(14) FIN EFFICIENCY           =    .523037
```

APPENDIX A

SI UNITS

Name	Quantity	Symbol
meter	length	m
square meter	area	m^2
cubic meter	volume	m^3
kilogram	mass	kg
second	time	s
Newton	force	N
degree Kelvin	temperature	^{o}K
degree Celsius	temperature	^{o}C
meter/second	velocity	m/s
newton/square meter	pressure	N/m^2
joule	work	J
joule/second (watt)	power	J/s (W)

APPENDIX B

ENGLISH UNITS

Name	Quantity	Symbol
foot	length	ft
square foot	area	ft^2
cubic foot	volume	ft^3
pound	mass	lb
second	time	s
pound force	force	lbf
degree Fahrenheit	temperature	^oF
degree Rankine (absolute)	temperature	^oR
foot/second	velocity	ft/s
ft/square foot	pressure	lbf/m^2
British Thermal Unit	work	BTU
foot-pound force	work	ft-lbf
British Thermal Unit/sec	power	BTU/s
horsepower	power	hp
foot-pound force/second	power	ft-lbf/s

APPENDIX C

PROGRAM LISTING

```
C*********************************************************************
C
C INDEX:  USE THE THREE ALPHANUMERIC CHARACTERS BELOW AND THE SEARCH/
C         FIND FUNCTION OF YOUR EDITOR TO QUICKLY LOCATE THE DESIRED
C         SECTION.
C
C  C1A - HEADER
C  C1B - MAIN
C  C1C - INTRO MENUS
C  C1D - INPUT DATA
C  C1E - SUMMARIZE DATA INPUT
C  C1F - GOTO SUBROUTINES
C  C1G - SUMMARIZE DATA OUTPUT
C  C1H - FORMAT STATEMENTS
C  C2A - RTAN: HEADER1
C  C2B - RTAN: MAIN1
C  C2C - RTAN: FCN1
C  C2D - RTAN: RKFSY1
C  C2E - RTAN: DERIV1
C  C2F - RTAN: MDLIN1
C  C3A - RTSY: HEADER2
C  C3B - RTSY: MAIN2
C  C3C - RTSY: FCN2
C  C3D - RTSY: RKFSY2
C  C3E - RTSY: DERIV2
C  C3F - RTSY: MDLIN2
C  C4A - TPAN: HEADER3
C  C4B - TPAN: MAIN3
C  C4C - TPAN: FCN3
C  C4D - TPAN: RKFSY3
C  C4E - TPAN: DERIV3
C  C4F - TPAN: MDLIN3
C  C5A - TPSY: HEADER4
C  C5B - TPSY: MAIN4
C  C5C - TPSY: FCN4
C  C5D - TPSY: RKFSY4
C  C5E - TPSY: DERIV4
C  C5F - TPSY: MDLIN4
C  C6A - TRAN: HEADER5
C  C6B - TRAN: MAIN5
C  C6C - TRAN: FCN5
C  C6D - TRAN: RKFSY5
C  C6E - TRAN: DERIV5
C  C6F - TRAN: MDLIN5
C  C7A - TRSY: HEADER6
C  C7B - TRSY: MAIN6
```

```
C  C7C - TRSY: FCN6
C  C7D - TRSY: RKFSY6
C  C7E - TRSY: DERIV6
C  C7F - TRSY: MDLIN6
C  C8A - SI
C  C9A - ENG
C
C1A ***** HEADER *****************************************************
C
C  MAIN PROGRAM: FIN.FOR
C
C  PROGRAMMER:  D. R. JOHNSON
C
C  DATE: 10 JUN 90
C
C-----------------------------------------------------------------------
C
C  PURPOSE:  THIS PROGRAM SOLVES THE FOLLOWING LONGITUDINAL FIN
C            HEAT TRANSFER PROBLEMS:
C
C  FIN PROFILES:  RECTANGULAR,TRAPEZOIDAL, & TRIANGULAR
C
C  PROBLEM TYPES:  ANALYSIS & SYNTHESIS
C
C  ENVIRONMENTS:  FREE SPACE & NON-FREE SPACE
C
C-----------------------------------------------------------------------
C
C  ANALYSIS PROBLEM:
C
C  INPUTS:
C
C   RECTANGLE            TRAPEZOID            TRIANGLE
C   ----------------     ----------------     ----------------
C   BASE TEMPERATURE     BASE TEMPERATURE     BASE TEMPERATURE
C   LENGTH               LENGTH               LENGTH
C   BASE THICKNESS       BASE THICKNESS       BASE THICKNESS
C                        TIP THICKNESS        *
C   CONDUCTIVITY         CONDUCTIVTY          CONDUCTIVITY
C   ABSORPTIVITY         ABSORPTIVITY         ABSORPTIVITY
C   EMISSIVITY           EMISSIVITY           EMISSIVITY
C   HEIGHT               HEIGHT               HEIGHT
C   EXTERNAL HEAT FLUX   EXTERNAL HEAT FLUX   EXTERNAL HEAT FLUX
C
C   *  TIP/BASE THICKNESS RATIO ASSUMED = 0.01
C
C  OUTPUTS:
C
C   RECTANGLE            TRAPEZOID            TRIANGLE
C   ----------------     ----------------     ----------------
C   HEAT DISSIPATED      HEAT DISSIPATED      HEAT DISSIPATED
C   TIP TEMPERATURE      TIP TEMPERATURE      TIP TEMPERATURE
C   EFFICIENCY           EFFICIENCY           EFFICIENCY
```

```
C
C----------------------------------------------------------------------
C
C  SYNTHESIS PROBLEM:
C
C  INPUTS:
C
C   RECTANGLE            TRAPEZOID            TRIANGLE
C   ----------------     ----------------     ----------------
C   BASE TEMPERATURE     BASE TEMPERATURE     BASE TEMPERATURE
C   LENGTH               LENGTH               LENGTH
C   BASE THICKNESS       BASE THICKNESS       BASE THICKNESS
C   TIP THICKNESS        *
C   CONDUCTIVITY    CONDUCTIVTY     CONDUCTIVITY
C   ABSORPTIVITY    ABSORPTIVITY        ABSORPTIVITY
C   EMISSIVITY      EMISSIVITY      EMISSIVITY
C   EXTERNAL HEAT FLUX  EXTERNAL HEAT FLUX  EXTERNAL HEAT FLUX
C   HEAT INPUT      HEAT INPUT      HEAT INPUT
C
C   *  TIP/BASE THICKNESS RATIO ASSUMED = 0.01
C
C  OUTPUTS:
C
C   RECTANGLE            TRAPEZOID            TRIANGLE
C   ----------------     ----------------     ----------------
C   HEIGHT               HEIGHT               HEIGHT
C   TIP TEMPERATURE      TIP TEMPERATURE      TIP TEMPERATURE
C   EFFICIENCY           EFFICIENCY           EFFICIENCY
C
C----------------------------------------------------------------------
C
C  PARAMETERS:
C
C  TB - TEMPERATURE AT THE BASE OF THE FIN
C  TE - TEMPERATURE AT THE TIP OF THE FIN
C  HT - HEIGHT OF THE FIN
C  L - LENGTH OF THE FIN
C  DEL - THICKNESS OF FIN (RECTANGLULAR)
C  DEL0 - THICKNESS OF FIN AT BASE (TRAPEZOIDAL OR TRIANGULAR)
C  DELE - THICKNESS OF FIN AT TIP (TRAPEZOIDAL)
C  K - CONDUCTIVITY OF THE FIN
C  K1 - CONSTANT = 2 * SB * EMIS
C  K2 - CONSTANT = E * ABS
C  ABS - ABSORPTIVITY OF THE FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  Q - REAL HEAT DISSIPATED BY THE FIN
C  QI - IDEAL HEAT DISSIPATED BY THE FIN
C  SB - STEFAN-BOLTZMAN CONSTANT
C
C----------------------------------------------------------------------
C
```

```
C  SUBROUTINES:
C
C  RTAN - SOLVES RECTANGULAR FIN, ANALYSIS PROBLEM
C  RTSY - SOLVES RECTANGULAR FIN, SYNTHESIS PROBLEM
C  TPAN - SOLVES TRAPEZOIDAL FIN, ANALYSIS PROBLEM
C  TPSY - SOLVES TRAPEZOIDAL FIN, SYNTHESIS PROBLEM
C  TRAN - SOLVES TRIANGULAR FIN, ANALYSIS PROBLEM
C  TRSY - SOLVES TRIANGULAR FIN, SYNTHESIS PROBLEM
C
C-----------------------------------------------------------------------
C
C  UNITS:
C
C   TYPE          SI (INTERNATIONAL)   ENGLISH
C   --------      ------------------   -------
C   LINEAR        METERS               FEET
C   TEMPERATURE   CENTIGRADE           FAHRENHEIT
C   HEAT          WATTS                BTU/HR
C
C1B ***** MAIN *********************************************************
C
      REAL*8 TB,TE,HT,L,DEL,DEL0,DELE,K,ABS,EMIS,E,Q,QI,EFF
      INTEGER PASS,FLAG
      CHARACTER*2 ANS,UNITS,FIN,TYPE,MODE
      DATA TB,TE,HT,L,DEL,DEL0,DELE,K,ABS,EMIS,E,Q,QI,EFF/14*0.0D0/
      FLAG = 0
C
C-------- OPEN FILE FIN.DAT TO HOLD DATA FOR PRINTER -----------------
C
      OPEN(UNIT=9,
     &     FILE='FIN.DAT',
     &     ACCESS='SEQUENTIAL',
     &     FORM='FORMATTED',
     &     STATUS='UNKNOWN')
C
C1C ***** INTRODUCTION MENUS *******************************************
C
C-------- ENSURE IN "CAPS LOCK" MODE ---------------------------------
C
    1 PRINT 800
      READ 9000,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 1200
        GOTO 1
      ENDIF
      IF(ANS.EQ.'N') GOTO 1
C
C
C-------- SELECT MONITOR MODE ----------------------------------------
C
    2 PRINT 900
      READ 9000,MODE
      IF(MODE.NE.'1'.AND.MODE.NE.'2') THEN
```

```
        PRINT 1200
        GOTO 2
      ENDIF
C
C-------- SKIP MENUS ------------------------------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
    3 PRINT 1100
      READ 9000,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 1200
        GOTO 3
      ENDIF
      IF(ANS.EQ.'Y') GO TO 44
C
C-------- INTRO MENU - GENERAL --------------------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 1300
      PRINT 1301
   10 PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 10
      ENDIF
C
C-------- INTRO MENU - FIN ANALYSIS INPUT ---------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 1500
   15 PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 15
      ENDIF
C
C-------- INTRO MENU - FIN ANALYSIS OUTPUT --------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 1600
   20 PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 20
      ENDIF
C
```

```
C-------- INTRO MENU - FIN SYNTHESIS INPUT --------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 1700
   25 PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 25
      ENDIF
C
C-------- INTRO MENU - FIN SYNTHESIS OUTPUT -------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 1800
   30 PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 30
      ENDIF
C
C-------- INTRO MENU - UNITS MODE -----------------------------------
C
   35 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 1900
      PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 35
      ENDIF
C
C-------- INTRO MENU - NOTES ----------------------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 2000
      PRINT 2001
   40 PRINT 1400
      READ 9000,ANS
      IF(ANS.NE.' ') THEN
        PRINT 1200
        GOTO 40
      ENDIF
C
C1D ***** INPUT DATA ************************************************
C
C-------- SELECT UNITS MODE -----------------------------------------
C
```

```
   44 PASS = 0
   45 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   46 PRINT 2100
      READ 9000,UNITS
      IF(UNITS.NE.'1'.AND.UNITS.NE.'2') THEN
        PRINT 1200
        GOTO 46
      ENDIF
      IF(PASS.EQ.1) GOTO 60
C
C-------- SELECT FIN TYPE -------------------------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   50 PRINT 2200
      READ 9000,FIN
      IF(FIN.NE.'1'.AND.FIN.NE.'2'.AND.FIN.NE.'3') THEN
        PRINT 1200
        GOTO 50
      ENDIF
      IF(PASS.EQ.1.AND.FIN.EQ.'1') GOTO 70
      IF(PASS.EQ.1.AND.FIN.EQ.'2') GOTO 75
      IF(PASS.EQ.1.AND.FIN.EQ.'3') GOTO 80
C
C-------- SELECT PROBLEM TYPE ---------------------------------------
C
      IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   55 PRINT 2300
      READ 9000,TYPE
      IF(TYPE.NE.'1'.AND.TYPE.NE.'2') THEN
        PRINT 1200
        GOTO 55
      ENDIF
      IF(PASS.EQ.1.AND.TYPE.EQ.'1') GOTO 98
      IF(PASS.EQ.1.AND.TYPE.EQ.'2') GOTO 96
C
C-------- INPUT BASE TEMPERATURE ------------------------------------
C
   60 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   61 PRINT 3000
      READ *,TB
      IF(UNITS.EQ.'1'.AND.TB.LT.-273.15D0) THEN
        PRINT 3001
        PRINT 1001
        GOTO 61
      ENDIF
      IF(UNITS.EQ.'2'.AND.TB.LT.-460.0D0) THEN
        PRINT 3002
        PRINT 1001
        GOTO 61
```

```
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT FIN LENGTH -------------------------------------------
C
   65 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   66 PRINT 3100
      READ *,L
      IF(L.LE.0.0D0) THEN
        PRINT 3101
        PRINT 1001
        GOTO 66
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT BASE THICKNESS IF RECTANGLE --------------------------
C
   70 IF(FIN.EQ.'1') THEN
        IF(MODE.EQ.'1') PRINT 1000
        IF(MODE.EQ.'2') PRINT 1001
   71   PRINT 3200
        READ *,DEL
        IF(DEL.LE.0.0D0) THEN
          PRINT 3201
          PRINT 1001
          GOTO 71
        ENDIF
      ENDIF
      IF (PASS.EQ.1) GOTO 105
C
C-------- INPUT BASE AND TIP THICKNESSES IF TRAPEZOID -----------------
C
   75 IF(FIN.EQ.'2') THEN
        IF(MODE.EQ.'1') PRINT 1000
        IF(MODE.EQ.'2') PRINT 1001
   76   PRINT 3300
        READ *,DEL0,DELE
        IF(DEL0.LE.0.0D0.OR.DELE.LE.0.0D0) THEN
          PRINT 3301
          PRINT 1001
          GOTO 76
        ENDIF
      IF(DEL0.LE.DELE) THEN
          PRINT 3302
          PRINT 1001
          GOTO 76
        ENDIF
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT BASE THICKNESS IF TRIANGLE --------------------------
C
```

```
   80 IF(FIN.EQ.'3') THEN
        IF(MODE.EQ.'1') PRINT 1000
        IF(MODE.EQ.'2') PRINT 1001
   81   PRINT 3200
        READ *,DEL0
        IF(DEL0.LE.0.0D0) THEN
          PRINT 3201
          PRINT 1001
          GOTO 81
        ENDIF
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT CONDUCTIVITY -----------------------------------------
C
   85 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   86 PRINT 3400
      READ *,K
      IF(K.LE.0.0D0) THEN
        PRINT 3401
        PRINT 1001
        GOTO 86
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT ABSORPTIVITY -----------------------------------------
C
   90 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   91 PRINT 3500
      READ *,ABS
      IF(ABS.LT.0.0D0.OR.ABS.GT.1.0D0) THEN
        PRINT 3501
        PRINT 1001
        GOTO 91
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT EMISSIVITY -------------------------------------------
C
   92 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   93 PRINT 3600
      READ *,EMIS
      IF(EMIS.LT.0.0D0.OR.EMIS.GT.1.0D0) THEN
        PRINT 3601
        PRINT 1001
        GOTO 93
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT EXTERNAL HEAT FLUX -----------------------------------
```

```
C
   94 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
   95 PRINT 3700
      READ *,E
      IF(E.LT.0.0D0) THEN
        PRINT 3701
        PRINT 1001
        GOTO 95
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT HEAT ENTERING BASE IF SYNTHESIS PROBLEM --------------
C
   96 IF(TYPE.EQ.'2') THEN
        IF(MODE.EQ.'1') PRINT 1000
        IF(MODE.EQ.'2') PRINT 1001
   97   PRINT 3800
        READ *,Q
        IF(Q.LE.0.0D0) THEN
          PRINT 3801
          PRINT 1001
          GOTO 97
        ENDIF
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C-------- INPUT FIN HEIGHT IF ANALYSIS PROBLEM -----------------------
C
   98 IF(TYPE.EQ.'1') THEN
        IF(MODE.EQ.'1') PRINT 1000
        IF(MODE.EQ.'2') PRINT 1001
   99   PRINT 3900
        READ *,HT
        IF(HT.LE.0.0D0) THEN
          PRINT 3901
          PRINT 1001
          GOTO 99
        ENDIF
      ENDIF
      IF(PASS.EQ.1) GOTO 105
C
C1E ***** SUMMARIZE DATA INPUT ***************************************
C
C-------- PRINT INPUTS TO SCREEN ------------------------------------
C
  105 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 4000
C
C-------- PRINT UNITS TYPE ------------------------------------------
C
      IF(UNITS.EQ.'1') PRINT 4005
```

```
      IF(UNITS.EQ.'2') PRINT 4010
C
C-------- PRINT FIN TYPE ------------------------------------------
C
      IF(FIN.EQ.'1') PRINT 4015
      IF(FIN.EQ.'2') PRINT 4020
      IF(FIN.EQ.'3') PRINT 4025
C
C-------- PRINT PROBLEM TYPE --------------------------------------
C
      IF(TYPE.EQ.'1') PRINT 4030
      IF(TYPE.EQ.'2') PRINT 4035
C
C-------- PRINT COMMON INPUTS: BASE TEMP & LENGTH --------------------
C
      IF(UNITS.EQ.'1') PRINT 4040, TB,L
      IF(UNITS.EQ.'2') PRINT 4041, TB,L
C
C-------- IF RECTANGLE OR TRIANGLE PRINT BASE THICKNESS --------------
C
      IF(FIN.EQ.'1'.AND.UNITS.EQ.'1') PRINT 4045,DEL
      IF(FIN.EQ.'1'.AND.UNITS.EQ.'2') PRINT 4046,DEL
      IF(FIN.EQ.'3'.AND.UNITS.EQ.'1') PRINT 4045,DEL0
      IF(FIN.EQ.'3'.AND.UNITS.EQ.'2') PRINT 4046,DEL0
C
C-------- IF TRAPEZOID PRINT BASE & TIP THICKNESS --------------------
C
      IF(FIN.EQ.'2'.AND.UNITS.EQ.'1') PRINT 4050,DEL0,DELE
      IF(FIN.EQ.'2'.AND.UNITS.EQ.'2') PRINT 4051,DEL0,DELE
C
C-------- PRINT COMMON INPUTS: CONDUCTIVITY, ABSORPTIVITY ------------
C         EMISSIVITY, AND EXTERNAL FLUX
C
      IF(UNITS.EQ.'1') PRINT 4055,K,ABS,EMIS,E
      IF(UNITS.EQ.'2') PRINT 4056,K,ABS,EMIS,E
C
C-------- IF SYNTHESIS PROBLEM PRINT HEAT INPUT THRU BASE ------------
C
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'1') PRINT 4060,Q
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'2') PRINT 4061,Q
C
C-------- IF ANALYSIS PROBLEM PRINT FIN HEIGHT -----------------------
C
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'1') PRINT 4065,HT
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'2') PRINT 4066,HT
C
C-------- CHECK TO SEE IF INPUTS ARE CORRECT -------------------------
C
  110 PRINT 4700
      READ 9000,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 1200
        GOTO 110
```

```
      ENDIF
      IF(ANS.EQ.'Y') GOTO 115
C
C-------- CORRECT ANY WRONG INPUTS ----------------------------------
C
      PASS = 1
      PRINT 4750
      READ 9000,ANS
      IF(ANS.EQ.'1')                    GOTO 44
      IF(ANS.EQ.'2')                    GOTO 50
      IF(ANS.EQ.'3')                    GOTO 55
      IF(ANS.EQ.'4')                    GOTO 60
      IF(ANS.EQ.'5')                    GOTO 65
      IF(ANS.EQ.'6'.AND.FIN.EQ.'1')     GOTO 70
      IF(ANS.EQ.'6'.AND.FIN.EQ.'2')     GOTO 75
      IF(ANS.EQ.'6'.AND.FIN.EQ.'3')     GOTO 80
      IF(ANS.EQ.'7')                    GOTO 85
      IF(ANS.EQ.'8')                    GOTO 90
      IF(ANS.EQ.'9')                    GOTO 92
      IF(ANS.EQ.'10')                   GOTO 94
      IF(ANS.EQ.'11'.AND.TYPE.EQ.'2') GOTO 96
      IF(ANS.EQ.'11'.AND.TYPE.EQ.'1') GOTO 98
      PRINT 1200
      GOTO 105
C
C1F ***** GOTO SUBROUTINES *********************************************
C
C-------- FOR ENGLISH: CONVERT ALL TO SI ------------------------------
C
  115 IF(UNITS.EQ.'2') CALL SI(TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q)
C
C-------- CONVERT C TO K --------------------------------------------
C
      TB = TB + 273.15D0
C
C-------- SELECT SUBROUTINE -----------------------------------------
C
      IF(FIN.EQ.'1'.AND.TYPE.EQ.'1') CALL RTAN(TB,TE,L,HT,DEL,K,ABS,
     &   EMIS,E,QI,Q,EFF,FLAG)
C
      IF(FIN.EQ.'1'.AND.TYPE.EQ.'2') CALL RTSY(TB,TE,L,HT,DEL,K,ABS,
     &   EMIS,E,QI,Q,EFF,FLAG)
C
      IF(FIN.EQ.'2'.AND.TYPE.EQ.'1') CALL TPAN(TB,TE,L,HT,DEL0,DELE,
     &   K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
      IF(FIN.EQ.'2'.AND.TYPE.EQ.'2') CALL TPSY(TB,TE,L,HT,DEL0,DELE,
     &   K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
      IF(FIN.EQ.'3'.AND.TYPE.EQ.'1') CALL TRAN(TB,TE,L,HT,DEL0,DELE,
     &   K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
      IF(FIN.EQ.'3'.AND.TYPE.EQ.'2') CALL TRSY(TB,TE,L,HT,DEL0,DELE,
```

```
     &   K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
C1G ***** SUMMARIZE DATA OUTPUT **************************************
C
C-------- IF THERE IS NO SOLUTION, START OVER WITH NEW PROBLEM ------
C
      IF(FLAG.EQ.1) GOTO 125
C
C-------- CONVERT K TO C --------------------------------------------
C
      TB = TB - 273.15D0
      TE = TE - 273.15D0
C
C-------- FOR ENGLISH: COVERT ALL BACK TO ENGLISH ------------------
C
      IF(UNITS.EQ.'2') CALL ENG(TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q)
C
C-------- REPEAT INPUTS ---------------------------------------------
C
  116 IF(MODE.EQ.'1') PRINT 1000
      IF(MODE.EQ.'2') PRINT 1001
      PRINT 4800
C
C-------- PRINT UNITS TYPE ------------------------------------------
C
      IF(UNITS.EQ.'1') PRINT 4005
      IF(UNITS.EQ.'2') PRINT 4010
C
C-------- PRINT FIN TYPE --------------------------------------------
C
      IF(FIN.EQ.'1') PRINT 4015
      IF(FIN.EQ.'2') PRINT 4020
      IF(FIN.EQ.'3') PRINT 4025
C
C-------- PRINT PROBLEM TYPE ----------------------------------------
C
      IF(TYPE.EQ.'1') PRINT 4030
      IF(TYPE.EQ.'2') PRINT 4035
C
C-------- PRINT COMMON INPUTS: BASE TEMP & LENGTH -------------------
C
      IF(UNITS.EQ.'1') PRINT 4040,TB,L
      IF(UNITS.EQ.'2') PRINT 4041,TB,L
C
C-------- IF RECTANGLE OR TRIANGLE PRINT BASE THICKNESS --------------
C
      IF(FIN.EQ.'1'.AND.UNITS.EQ.'1') PRINT 4045,DEL
      IF(FIN.EQ.'1'.AND.UNITS.EQ.'2') PRINT 4046,DEL
      IF(FIN.EQ.'3'.AND.UNITS.EQ.'1') PRINT 4045,DEL0
      IF(FIN.EQ.'3'.AND.UNITS.EQ.'2') PRINT 4046,DEL0
C
C-------- IF TRAPEZOID PRINT BASE & TIP THICKNESS -------------------
C
```

```
      IF(FIN.EQ.'2'.AND.UNITS.EQ.'1') PRINT 4050,DEL0,DELE
      IF(FIN.EQ.'2'.AND.UNITS.EQ.'2') PRINT 4051,DEL0,DELE
C
C-------- PRINT COMMON INPUTS: CONDUCTIVITY, ABSORPTIVITY -----------
C         EMISSIVITY, AND EXTERNAL FLUX
C
      IF(UNITS.EQ.'1') PRINT 4055,K,ABS,EMIS,E
      IF(UNITS.EQ.'2') PRINT 4056,K,ABS,EMIS,E
C
C-------- IF SYNTHESIS PROBLEM, PRINT HEAT INPUT ---------------------
C
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'1') PRINT 4060,Q
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'2') PRINT 4061,Q
C
C-------- IF ANALYSIS PROBLEM, PRINT FIN HEIGHT ---------------------
C
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'1') PRINT 4065,HT
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'2') PRINT 4066,HT
C
C-------- SUMMARIZE OUTPUT ------------------------------------------
C
      PRINT 4805
C
C-------- IF ANALYSIS PROBLEM, PRINT HEAT OUTPUT --------------------
C
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'1') PRINT 4062,Q
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'2') PRINT 4063,Q
      IF(TYPE.EQ.'1'.AND.Q.LE.0.0D0) PRINT 4064
C
C-------- IF SYNTHESIS PROBLEM, PRINT FIN HEIGHT --------------------
C
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'1') PRINT 4067,HT
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'2') PRINT 4068,HT
      IF(TYPE.EQ.'2'.AND.HT.LE.0.0D0) PRINT 4069
C
C-------- PRINT TIP TEMPERATURE -------------------------------------
C
      IF(UNITS.EQ.'1') PRINT 4810,TE
      IF(UNITS.EQ.'2') PRINT 4811,TE
      IF(UNITS.EQ.'1'.AND.TE.LT.-273.15D0) PRINT 4812
      IF(UNITS.EQ.'2'.AND.TE.LT.-460.0D0) PRINT 4813
C
C-------- PRINT FIN EFFICIENCY --------------------------------------
C
      PRINT 4815,EFF
      IF(EFF.LT.0.0D0.OR.EFF.GT.1.0D0) PRINT 4816
C
C-------- ALSO WRITE OUTPUTS TO FILE --------------------------------
C
                                         WRITE(9,4800)
      IF(UNITS.EQ.'1')                   WRITE(9,4005)
      IF(UNITS.EQ.'2')                   WRITE(9,4010)
      IF(FIN.EQ.'1')                     WRITE(9,4015)
```

```
      IF(FIN.EQ.'2')                          WRITE(9,4020)
      IF(FIN.EQ.'3')                          WRITE(9,4025)
      IF(TYPE.EQ.'1')                         WRITE(9,4030)
      IF(TYPE.EQ.'2')                         WRITE(9,4035)
      IF(UNITS.EQ.'1')                        WRITE(9,4040) TB,L
      IF(UNITS.EQ.'2')                        WRITE(9,4041) TB,L
      IF(FIN.EQ.'1'.AND.UNITS.EQ.'1')         WRITE(9,4045) DEL
      IF(FIN.EQ.'1'.AND.UNITS.EQ.'2')         WRITE(9,4046) DEL
      IF(FIN.EQ.'3'.AND.UNITS.EQ.'1')         WRITE(9,4045) DEL0
      IF(FIN.EQ.'3'.AND.UNITS.EQ.'2')         WRITE(9,4046) DEL0
      IF(FIN.EQ.'2'.AND.UNITS.EQ.'1')         WRITE(9,4050) DEL0,DELE
      IF(FIN.EQ.'2'.AND.UNITS.EQ.'2')         WRITE(9,4051) DEL0,DELE
      IF(UNITS.EQ.'1')                   WRITE(9,4055) K,ABS,EMIS,E
      IF(UNITS.EQ.'2')                   WRITE(9,4056) K,ABS,EMIS,E
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'1')        WRITE(9,4060) Q
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'2')        WRITE(9,4061) Q
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'1')        WRITE(9,4065) HT
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'2')        WRITE(9,4066) HT
                                              WRITE(9,4805)
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'1')        WRITE(9,4062) Q
      IF(TYPE.EQ.'1'.AND.UNITS.EQ.'2')        WRITE(9,4063) Q
      IF(Q.LE.0.0D0)                          WRITE(9,4064)
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'1')        WRITE(9,4067) HT
      IF(TYPE.EQ.'2'.AND.UNITS.EQ.'2')        WRITE(9,4068) HT
      IF(HT.LE.0.0D0)                         WRITE(9,4069)
      IF(UNITS.EQ.'1')                        WRITE(9,4810) TE
      IF(UNITS.EQ.'2')                        WRITE(9,4811) TE
      IF(UNITS.EQ.'1'.AND.TE.LT.-273.15D0)    WRITE(9,4812)
      IF(UNITS.EQ.'2'.AND.TE.LT.-460.0D0)     WRITE(9,4812)
                                              WRITE(9,4815) EFF
C
C-------- CONVERT OUTPUT TO OTHER UNITS ? --------------------------
C
 119  PRINT 5199
      READ 9000,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 1200
        GOTO 119
      ENDIF
      IF(ANS.EQ.'N') GOTO 120
      IF(UNITS.EQ.'1') THEN
        CALL ENG(TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q)
        UNITS = '2'
        GOTO 116
      ENDIF
      IF(UNITS.EQ.'2') THEN
        CALL SI(TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q)
        UNITS = '1'
        GOTO 116
      ENDIF
C
C-------- DO SAME PROBLEM AGAIN WITH DIFFERENT INPUTS ? ------------
C
```

```
 120  PRINT 5200
      READ 9000,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 1200
        GOTO 120
      ENDIF
      IF(ANS.EQ.'Y') GOTO 105
C
C-------- NEW PROBLEM ? ---------------------------------------------
C
 125  PRINT 5300
      READ 9000,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 1200
        GOTO 125
      ENDIF
      IF(ANS.EQ.'Y') GO TO 44
      CLOSE(UNIT=9)
      STOP
C
C1H ***** FORMAT STATEMENTS ******************************************
C
 800  FORMAT(/,' ARE YOU IN "CAPS LOCK" MODE (Y OR N) ?            ')
 900  FORMAT(/,' SELECT VIEWING MODE (1 OR 2):                    ',/,
     & ' 1.  SCREEN VIEWING (SCREEN IS CLEARED BEFORE NEXT DISPLAY) ',/,
     & ' 2.  LINE VIEWING (OUTPUT TO SCREEN IS LINE BY LINE)       ')
1000  FORMAT(24(/))
1001  FORMAT(/)
1100  FORMAT(/,' DO YOU WISH TO SKIP THE INTRODUCTION AND PROCEED'     ,
     & ' DIRECTLY TO A PROBLEM',/,' (Y OR N) ?                       ')
1200  FORMAT(/,' *************** INCORRECT ENTRY! ***************   ',/)
1300  FORMAT(/,'                INTRODUCTION                          ',/,
     & '                                                           ',/,
     & ' THE PURPOSE OF THIS PROGRAM IS TO ANALYZE AND SYNTHESIZE   ',/,
     & ' RADIATIVE HEAT TRANSFER IN LONGITUDINAL FINS OF THREE      ',/,
     & ' PROFILES:                                                  ',/,
     & '                                                           ',/,
     & ' 1.  RECTANGULAR                                            ',/,
     & ' 2.  TRAPEZOIDAL                                            ',/,
     & ' 3.  TRIANGULAR                                             ',/,
     & '                                                           ',/,
     & ' TWO TYPES OF PROBLEMS ARE SOLVED:                          ',/,
     & '                                                           ',/,
     & ' 1.  ANALYSIS                                               ',/,
     & ' 2.  SYNTHESIS                                              ')
1301  FORMAT(/,' AND TWO ENVIRONMENTS ARE CONSIDERED:                ',/,
     & '                                                           ',/,
     & ' 1.  FREE SPACE                                             ',/,
     & ' 2.  NON-FREE SPACE                                         ')
1400  FORMAT(/,' ---------- PRESS RETURN TO CONTINUE ----------     ')
1500  FORMAT(/,' FOR THE FIN ANALYSIS PROBLEM THE FOLLOWING INPUTS'  ,/,
     & ' ARE NEEDED:                                                ',/,
     & '                                                           ',/,
```

```
     & ' RECTANGLE          TRAPEZOID          TRIANGLE             ',/,
     & ' ----------------   ----------------   ----------------     ',/,
     & ' BASE TEMPERATURE   BASE TEMPERATURE   BASE TEMPERATURE     ',/,
     & ' LENGTH             LENGTH             LENGTH               ',/,
     & ' BASE THICKNESS     BASE THICKNESS     BASE THICKNESS       ',/,
     & '                    TIP THICKNESS      *                    ',/,
     & ' ABSORPTIVITY       ABSORPTIVITY       ABSORPTVITY          ',/,
     & ' EMISSIVITY         EMISSIVITY         EMISSIVITY           ',/,
     & ' CONDUCTIVITY       CONDUCTIVITY       CONDUCTIVITY         ',/,
     & ' HEIGHT             HEIGHT             HEIGHT               ',/,
     & ' INCIDENT FLUX      INCIDENT FLUX      INCIDENT FLUX        ',/,
     & '                                                            ',/,
     & ' *  TIP/BASE THICKNESS RATIO ASSUMED = 0.01                  ')
1600 FORMAT(/,' THE FIN ANALYSIS OUTPUTS ARE:                        ',/,
     & '                                                            ',/,
     & ' RECTANGLE          TRAPEZOID           TRIANGLE            ',/,
     & ' ----------------   ----------------   ----------------     ',/,
     & ' HEAT DISSIPATED    HEAT DISSIPATED    HEAT DISSIPATED      ',/,
     & ' TIP TEMPERATURE    TIP TEMPERATURE    TIP TEMPERATURE      ',/,
     & ' EFFICIENCY         EFFICIENCY         EFFICIENCY            ')
1700 FORMAT(/,' FOR THE FIN SYNTHESIS PROBLEM THE FOLLOWING INPUTS' ,/,
     & ' ARE NEEDED:                                                ',/,
     & '                                                            ',/,
     & ' RECTANGLE            TRAPEZOID         TRIANGLE            ',/,
     & ' ----------------   ----------------   ----------------     ',/,
     & ' BASE TEMPERATURE   BASE TEMPERATURE   BASE TEMPERATURE     ',/,
     & ' LENGTH             LENGTH             LENGTH               ',/,
     & ' BASE THICKNESS     BASE THICKNESS     BASE THICKNESS       ',/,
     & '                    TIP THICKNESS      *                    ',/,
     & ' CONDUCTIVITY       CONDUCTIVITY       CONDUCTIVITY         ',/,
     & ' ABSORPTIVITY       ABSORPTIVITY       ABSORPTIVITY         ',/,
     & ' EMISSIVITY         EMISSIVITY         EMISSIVITY           ',/,
     & ' HEAT INPUT         HEAT INPUT         HEAT INPUT           ',/,
     & '                                                            ',/,
     & ' *  TIP/BASE THICKNESS ASSUMED = 0.01                        ')
1800 FORMAT(/,' THE FIN SYNTHESIS OUTPUTS ARE:                       ',/,
     & '                                                            ',/,
     & ' RECTANGLE          TRAPEZOID          TRIANGLE             ',/,
     & ' ----------------   ----------------   ----------------     ',/,
     & ' HEIGHT             HEIGHT             HEIGHT               ',/,
     & ' TIP TEMPERATURE    TIP TEMPERATURE    TIP TEMPERATURE      ',/,
     & ' EFFICIENCY         EFFICIENCY         EFFICIENCY            ')
1900 FORMAT(/,' THERE ARE TWO MEASUREMENTS MODES:                    ',/,
     & '                                                            ',/,
     & ' 1.   SI UNITS                                              ',/,
     & ' 2.   ENGLISH UNITS                                          ')
2000 FORMAT(/,' NOTES:                                               ',/,
     & '                                                            ',/,
     & ' 1.   MUST BE IN "CAPS LOCK" MODE.                          ',/,
     & ' 2.   TO ESCAPE, PRESS "CONTROL BREAK" AND REENTER PGM.     ',/,
     & ' 3.   PROGRAM INTERRUPTION BY "UNDERFLOW","OVERFLOW" OR     ',/,
     & '      OTHER PROCESSOR ERRORS INDICATES PROBLEMS IN THE      ',/,
     & '      INPUT DATA - CHECK FOR CORRECTNESS.                   ',/,
```

```
     & ' 4.  THE MAXIMUM NUMBER OF INTERVALS SEARCHED IS 100. THIS  ',/,
     & '     CAN BE INCREASED IN INCREMENTS OF 100 BY MENU CHOICE.  ',/,
     & '     THE MAXIMUM NUMBER OF ITERATIONS FOR FINDING THE ROOT  ',/,
     & '     IS 50.  THIS CAN ALSO BE INCREASED IN INCREMENTS OF    ',/,
     & '     50.  MOST PROBLEMS WILL CONVERGE WITHIN THESE LIMITS.  ',/,
     & '     HOWEVER SOME INPUTS CAUSE DIVERGENCE OR OSCILLATION.   ',/,
     & ' 5.  ALTHOUGH SIX DECIMAL PLACES ARE SHOWN IN THE INPUT/    ',/,
     & '     OUTPUT DATA SUMMARIES, IT IS NOT GUARANTEED. IT        ',/,
     & '     DEPENDS ON THE ACCURACY OF INPUT. THE BEST THAT        ',/,
     & '     CAN BE EXPECTED IN SINGLE PRECISION IS 5.               ')
2001 FORMAT(' 6. OUTPUT IS PRINTED TO THE SCREEN AND TO A FILE      ',/,
     & '     CALLED "FIN.DAT".  UPON EXITING THE PROGRAM THIS       ',/,
     & '     CAN BE EDITED WITH YOUR EDITOR OR WORD PROCESSOR       ',/,
     & '     OR CAN BE PRINTED DIRECTLY WITH THE DOS COMMAND        ',/,
     & '     "PRINT FIN.DAT".  WARNING:  THIS FILE IS WRITTEN       ',/,
     & '     OVER WITH EACH BEGINNING OF THE PROGRAM!                ')
2100 FORMAT(/,' SELECT THE APPROPRIATE UNITS (1 OR 2):              ',/,
     & '                                                            ',/,
     & ' 1.  SI UNITS (METERS,DEGREES CENTIGRADE,WATT,ETC.)         ',/,
     & ' 2.  ENGLISH UNITS (FEET,DEGREES FAHRENHEIT,BTU,ETC.)        ')
2200 FORMAT(/,' SELECT THE TYPE OF FIN (1, 2, OR 3):                ',/,
     & '                                                            ',/,
     & ' 1.  RECTANGULAR                                            ',/,
     & ' 2.  TRAPEZOIDAL                                            ',/,
     & ' 3.  TRIANGULAR                                              ')
2300 FORMAT(/,' SELECT THE TYPE OF PROBLEM (1 OR 2):                ',/,
     & '                                                            ',/,
     & ' 1.  ANALYSIS                                               ',/,
     & ' 2.  SYNTHESIS                                              ',/,
     & '                                                             ')
3000 FORMAT(' WHAT IS THE BASE TEMPERATURE (DEGREES C OR F) ?       ',/,
     & ' EXAMPLE:  440   234.56   0.3504567E3                        ')
3001 FORMAT(/,' **** ERROR:  TEMPERATURE MUST BE GREATER THAN OR'     ,
     & ' EQUAL TO ABSOLUTE'                                         ,/,
     &         '                  ZERO (-273.15 DEGREES C) !         ')
3002 FORMAT(/,' **** ERROR:  TEMPERATURE MUST BE GREATER THAN OR'     ,
     & ' EQUAL TO ABSOLUTE'                                         ,/,
     &         '              ZERO (-460 DEGREES F) !                ')
3100 FORMAT(' WHAT IS THE LENGTH OF THE FIN (M OR FT) ?             ',/,
     & ' EXAMPLE:  1     3.4565   0.73456E1                          ')
3101 FORMAT(/,' **** ERROR:  LENGTH MUST BE GREATER THAN ZERO !      ')
3200 FORMAT(' WHAT IS THE FIN THICKNESS AT THE BASE (M OR FT) ?     ',/,
     & ' EXAMPLE: .0095   0.0034   0.7345645-2                       ')
3201 FORMAT(/,' **** ERROR:  BASE THICKNESS MUST BE GREATER THAN'     ,
     & ' ZERO !'                                                     )
3300 FORMAT(' WHAT IS THE FIN THICKNESS AT THE BASE AND TIP ?'       ,
     & ' (M OR FT) ?',/,' ENTER 2 NUMBERS SEPARATED BY A SPACE.'    ,/,
     & ' EXAMPLE: .034 0.0068   OR   0.34E-2 0.234567E-2             ')
3301 FORMAT(/,' **** ERROR:  THICKNESS MUST BE GREATER THAN ZERO !   ')
3302 FORMAT(/,' **** ERROR:  BASE THICKNESS MUST BE GREATER THAN'     ,
     & ' TIP THICKNESS !'                                            )
3400 FORMAT(' WHAT IS THE CONDUCTIVITY (W/M-C OR BTU/FT-HR-F) ?     ',/,
     & ' EXAMPLE: 154    33.06    0.2563455E+3                       ')
```

```
3401  FORMAT(/,' **** ERROR:  CONDUCTIVITY MUST BE GREATER ZERO !     ')
3500  FORMAT(' WHAT IS THE ABSORPTIVITY (DIMENSIONLESS) ?'             ,/,
     & ' EXAMPLE: .12     0.56     2.5E-1                               ')
3501  FORMAT(/,' **** ERROR:  ABSORPTIVITY MUST BE GREATER THAN OR'     ,
     & ' EQUAL TO ZERO'                                                ,/,
     &          '                AND LESS THAN OR EQUAL TO ONE !        ')
3600  FORMAT(' WHAT IS THE EMISSIVITY (DIMENSIONLESS) ?               ',/,
     & ' EXAMPLE: .4      0.55     75.0E-2                              ')
3601  FORMAT(/,' **** ERROR:  EMISSIVITY MUST BE GREATER THAN OR'       ,
     & ' EQUAL TO ZERO'                                                ,/,
     &          '                AND LESS THAN OR EQUAL TO ONE !        ')
3700  FORMAT(' WHAT IS THE EXTERNAL HEAT FLUX (W OR BTU/HR) ?         ',/,
     & ' EXAMPLE: 675     450.78   0.234567+3                           ')
3701  FORMAT(/,' **** ERROR:  EXTERNAL HEAT FLUX MUST BE GREATER'       ,
     & ' THAN OR EQUAL TO ZERO !'                                       )
3800  FORMAT(' WHAT IS THE HEAT ENTERING THE BASE (W OR BTU/HR) ?   ',/,
     & ' EXAMPLE: 555     204.56   0.750567+3                           ')
3801  FORMAT(/,' **** ERROR:  HEAT INPUT THROUGH THE BASE MUST BE'      ,
     & ' GREATER THAN ZERO !'                                           )
3900  FORMAT(' WHAT IS THE HEIGHT OF THE FIN (M OR FT) ?              ',/,
     & ' EXAMPLE: .45     0.95     7.5E-1                               ')
3901  FORMAT(/,' **** ERROR:  FIN HEIGHT MUST BE GREATER THAN ZERO !  ')
4000  FORMAT(/,' INPUT DATA SUMMARY:                                  ',/,
     & '                                                                ')
4005  FORMAT('  (1) UNITS                 =    SI                       ')
4010  FORMAT('  (1) UNITS                 =    ENGLISH                  ')
4015  FORMAT('  (2) FIN                   =    RECTANGLE                ')
4020  FORMAT('  (2) FIN                   =    TRAPEZOID                ')
4025  FORMAT('  (2) FIN                   =    TRIANGLE                 ')
4030  FORMAT('  (3) PROBLEM               =    ANALYSIS                 ')
4035  FORMAT('  (3) PROBLEM               =    SYNTHESIS                ')
4040  FORMAT('  (4) BASE TEMPERATURE      =',G15.6,' C                ',/,
     &      '  (5) FIN LENGTH            =',G15.6,' M                  ')
4045  FORMAT('  (6) BASE THICKNESS        =',G15.6,' M                  ')
4050  FORMAT('  (6) BASE/TIP THICKNESSES  =',G15.6,2X,G15.6,' M         ')
4055  FORMAT('  (7) CONDUCTIVITY          =',G15.6,' W/M-C            ',/,
     &      '  (8) ABSORPTIVITY          =',G15.6,                     /,
     &      '  (9) EMISSIVITY            =',G15.6,                     /,
     &      ' (10) EXTERNAL HEAT FLUX    =',G15.6,' W                  ')
4060  FORMAT(' (11) HEAT INPUT THRU BASE  =',G15.6,' W                  ')
4062  FORMAT(' (12) HEAT INPUT THRU BASE  =',G15.6,' W                  ')
4065  FORMAT(' (11) FIN HEIGHT            =',G15.6,' M                  ')
4067  FORMAT(' (12) FIN HEIGHT            =',G15.6,' M                  ')
4041  FORMAT('  (4) BASE TEMPERATURE      =',G15.6,' F                ',/,
     &      '  (5) FIN LENGTH            =',G15.6,' FT                 ')
4046  FORMAT('  (6) BASE THICKNESS        =',G15.6,' FT                 ')
4051  FORMAT('  (6) BASE/TIP THICKNESSES  =',G15.6,2X,G15.6,' FT        ')
4056  FORMAT('  (7) CONDUCTIVITY          =',G15.6,' BTU/HR-FT-F      ',/,
     &      '  (8) ABSORPTIVITY          =',G15.6,                     /,
     &      '  (9) EMISSIVITY            =',G15.6,                     /,
     &      ' (10) EXTERNAL HEAT INPUT   =',G15.6,' BTU/HR             ')
4061  FORMAT(' (11) HEAT INPUT THRU BASE  =',G15.6,' BTU/HR             ')
4063  FORMAT(' (12) HEAT INPUT THRU BASE  =',G15.6,' BTU/HR             ')
```

```
4064  FORMAT(' **** WARNING: HEAT INPUT THROUGH BASE LESS < 0,'          ,
     & ' RECHECK YOUR INPUTS !'                                         )
4066  FORMAT(' (11) FIN HEIGHT            =',G15.6,' FT                 ')
4068  FORMAT(' (12) FIN HEIGHT            =',G15.6,' FT                 ')
4069  FORMAT(' **** WARNING: FIN HEIGHT < OR = 0, RECHECK YOUR'            ,
     & ' INPUTS !'                                                       )
4700  FORMAT(/,' IS THIS CORRECT (Y OR N) ?                             ')
4750  FORMAT(/,' WHAT IS THE NUMBER OF THE INCORRECT ENTRY ?             ')
4800  FORMAT(/' INPUT/OUTPUT SUMMARY:                                   ',/,
     & '                                                                 ',/,
     & ' ********************** INPUT ***************************       '/)
4805  FORMAT(/                                          ,
     & ' ********************** OUTPUT *************************        '/)
4810  FORMAT(' (13) TIP TEMPERATURE       =',G15.6,' C                  ')
4811  FORMAT(' (13) TIP TEMPERATURE       =',G15.6,' F                  ')
4812  FORMAT(' **** WARNING: FIN TIP TEMP < -273.15 DEG C, RECHECK'  ,
     & ' YOUR INPUTS  !'                                                 )
4813  FORMAT(' **** WARNING: FIN TIP TEMP < -460.0 DEG F, RECHECK'      ,
     & ' YOUR INPUTS !'                                                  )
4815  FORMAT(' (14) FIN EFFICIENCY        =',G15.6                       )
4816  FORMAT(' **** WARNING: FIN EFFICIENCY < 0 OR > 1, RECHECK'        ,
     & ' YOUR INPUTS !'                                                  )
5199  FORMAT(/' WOULD YOU LIKE TO SEE OUPUT IN OTHER UNITS (Y OR N) ?' )
5200  FORMAT(/' WOULD YOU LIKE TO DO SAME PROBLEM AGAIN WITH'            ,
     & ' DIFFERENT INPUTS (Y OR N) ?'                                    )
5300  FORMAT(/' WOULD YOU LIKE TO START OVER WITH A NEW PROBLEM'         ,
     & ' (Y OR N)?'                                                      )
9000  FORMAT(A2)
      END
C
C
C2A ***** RTAN: HEADER1 *********************************************
C
C  SUBROUTINE RTAN(TB,TE,HT,L,DEL,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
C  FIN PROFILE: RECTANGLE
C
C  PROBLEM TYPE:  ANALYSIS
C
C  INPUT:  TB,HT,L,DEL,K,ABS,EMIS,E
C
C  OUTPUT: Q,TE,QI,EFF
C
C-------------------------------------------------------------------
C
C  PARAMETERS:
C
C  TB - TEMPERATURE AT THE BASE OF THE FIN
C  TE - TEMPERATURE AT THE TIP OF THE FIN
C  HT - HEIGHT OF THE FIN
C  L - LENGTH OF THE FIN
C  DEL - THICKNESS FIN
C  K - CONDUCTIVITY OF THE FIN
```

```
C  ABS - ABSORPTIVITY OF THE FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  Q - HEAT DISSIPATED BY THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  FLAG - 0 = CONVERGENCE, 1 = DIVERGENCE
C
C-----------------------------------------------------------------------
C
C  FUNCTIONS:
C
C  FCN1(X,TB,TE,HT,DEL,K,K1,K2):
C
C   FUNCTION WHOSE ROOT IS FOUND (BASE DERIVATIVE & TIP TEMP)
C
C-----------------------------------------------------------------------
C
C  SUBROUTINES:
C
C  MDLIN1(FCN1,X1,X2,XR,TB,HT,DEL,K,K1,K2):
C
C   FINDS THE ROOT OF FUNCTION FCN1 BY THE METHOD OF MODIFIED
C   LINEAR INTERPOLATION
C
C  RKFSY1(DERIV1,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL):
C
C   SOLVES A SECOND ORDER NONLINEAR DE BY RUNGE-KUTTE-FELHBERG
C   METHOD
C
C  DERIV1(T,F,DEL,K,K1,K2)
C
C   COMPUTES THE DERIVATIVES FOR RKFSY1
C
C2B ***** RTAN: MAIN1 *************************************************
C
      SUBROUTINE RTAN(TB,TE,L,HT,DEL,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
      REAL*8 TB,TE,L,HT,DEL,K,ABS,EMIS,E,QI,Q,EFF,X1,X2,XR,
     &       SB,K1,K2,FCN1,F1,F2,TBDER
      INTEGER I,J,PASS,FLAG
      CHARACTER*2 ANS
      EXTERNAL FCN1
C
C-------- DEFINE CONSTANTS ---------------------------------------------
C
      SB = 5.67051D-12
      K1 = 2.0D0*SB*EMIS
      K2 = E*ABS
C
C-------- SCALE INPUTS -------------------------------------------------
C
      HT = HT*100.0D0
      DEL = DEL*100.0D0
      L = L*100.0D0
```

```
      K = K*0.01D0
C
C-------- START COUNTERS -------------------------------------------
C
      PASS = 0
      I = 0
      J = 1
C
C-------- START SEARCH WITH INTERVAL [-1,0] ------------------------
C
      PRINT 200
      X1 = 0.0D0
      X2 = -1.0D0
      F1 = FCN1(X1,TB,TE,HT,DEL,K,K1,K2)
    5 I = I + 1
C
C-------- LIMIT SEARCH TO INCREMENTS OF 100 INTERVALS --------------
C
      IF(I.GT.J*100) GOTO 10
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,X1,F1,X2,F2
      F2 = FCN1(X2,TB,TE,HT,DEL,K,K1,K2)
C
C-------- CHECK FOR FUNCTION VALUES OPPOSITE IN SIGN ---------------
C
      IF(F1*F2.GE.0.0D0) THEN
        X1 = X2
        F1 = F2
        X2 = X2 - 1.0D0
        IF(PASS.GE.1) PASS = PASS + 1
        GOTO 5
      ENDIF
C
C-------- IF INTERVAL FOUND, GO FIND ROOT --------------------------
C
      PRINT 201
      GOTO 20
C
C-------- IF NO INTERVAL FOUND, CONTINUE SEARCH ? ------------------
C
   10 PRINT 400,J*100
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- IF UNABLE TO FIND INTERVAL, LOOK AT FUNCTION VALUES ? -------
```

```
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 15
      ENDIF
      IF(ANS.EQ.'N') THEN
    FLAG = 1
    RETURN
      ENDIF
      PASS = 1
      I = 0
      J = 1
      X1 = 0.0D0
      X2 = -1.0D0
      F1 = FCN1(X1,TB,TE,HT,DEL,K,K1,K2)
      GOTO 5
C
C-------- GET ROOT IN INTERVAL: DERIV AT BASE AND TEMP AT TIP -------
C
   20 CALL MDLIN1(FCN1,X1,X2,XR,TB,TE,DEL,K,K1,K2,HT)
      TBDER = XR
C
C-------- CALCULATE IDEAL HEAT DISSIPATED ---------------------------
C
      QI = 2.0D0*SB*EMIS*HT*L*TB**4.0D0
C
C-------- CALCULATE REAL HEAT DISSIPATED ----------------------------
C
      Q = -K*DEL*L*TBDER
C
C-------- CALCULATE FIN EFFICIENCY ----------------------------------
C
      EFF = Q/QI
C
C-------- SCALE OUTPUTS ---------------------------------------------
C
      HT = HT*0.01D0
      DEL = DEL*0.01D0
      L = L*0.01D0
      K = K*100.0D0
      RETURN
C
C-------- FORMAT ------------------------------------------------------
C
100   FORMAT('+','LOOKING IN INTERVAL: ',I3,'                      ')
200   FORMAT(/,' COMMENCING SEARCH FOR INTERVAL TO BRACKET ROOT.... ',/)
201   FORMAT(/,' INTERVAL FOUND, WILL NOW LOOK FOR ROOT ....        ',/)
300   FORMAT(' XL =',2X,D10.4,2X,'F(XL) =',2X,D10.4,2X,
     &       'XR =',2X,D10.4,2X,'F(XR) =',2X,D10.4)
400   FORMAT(/,' NO ROOT FOUND IN ',I3,' INTERVALS.  CONTINUE SEARCH'  ,
     & ' (Y OR N) ?'                                                   )
```

```
500   FORMAT(A2)
600   FORMAT(/,' **************** INCORRECT ENTRY ! ***************** ')
700   FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N)?     ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING. IF ',/,
     & ' F(XR) IS OF THE SAME SIGN AS F(XL) AND ITS ABSOLUTE VALUE  ',/,
     & ' IS INCREASING, THERE IS LITTLE PROBABILITY OF FINDING AN   ',/,
     & ' INTERVAL.  THE SAME GOES FOR VALUES OF F(XR) THAT OSCILLATE',/,
     & ' AROUND A CERTAIN VALUE.  FOR F(XR) TO BECOME OPPOSITE IN   ',/,
     & ' SIGN TO F(XR), IT MUST PASS THROUGH ZERO.                  ')
      END
C
C2C****** RTAN: FCN1(X,TB,TE,HT,DEL,K,K1,K2) ************************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE (BASE DERIV)
C  TB - BASE TEMP
C  TE - TIP TEMP
C  HT - FIN HEIGHT
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0D0*SB*EMIS
C  K2 - CONSTANT = DABS*E
C
C-------------------------------------------------------------------
C
      REAL*8 FUNCTION FCN1(X,TB,TE,HT,DEL,K,K1,K2)
      REAL*8 X,X0,XFINAL,T0(2),TEND(2),F(2),H,TB,TE,HT,DEL,
     &     K,K1,K2,TSAVE,TOL
      EXTERNAL DERIV1
C
C-------- INITIALIZE LEFT END BC'S: FCT VALUE KNOWN, GUESS DERIV ----
C
      T0(1)= TB
      T0(2)= X
C
C-------- INITIALIZE RIGHT END BC'S TO 0 ----------------------------
C
      TEND(1)=0.0D0
      TEND(2)=0.0D0
C
C-------- DEFINE PARAMETERS -----------------------------------------
C
      TSAVE = 1000.0D0
      H = 0.1D0
      TOL = 0.0001D0
      X0 = 0.0D0
      XFINAL = HT
   10 IF (X0 .LT. XFINAL+0.0001D0) THEN
       CALL RKFSY1(DERIV1,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL)
C
C-------- INCREASING T MEANS DIVERGENCE -----------------------------
C
```

```
        IF (TEND(1).GE.TSAVE) GO TO 20
        T0(1) = TEND(1)
        TSAVE = TEND(1)
        T0(2) = TEND(2)
        GOTO 10
       ENDIF
   20  CONTINUE
       FCN1 = TEND(2)
       TE = TEND(1)
       RETURN
       END
C
C2D ***** RTAN: RKFSY1(DERIV1,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL)
C
C  PARAMTERS:
C
C  RKFSY1 - SUBROUTINE THAT SOLVES A SYSTEM OF 2 FIRST ORDER
C           DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-FEHLBERG
C           METHOD. THE EQUATIONS ARE OF THE FORM:
C
C           DT/DX = Y = F1(X,T)
C           DY/DX = F2(X,T,Y)
C
C  DERIV1 - A SUBROUTINE THAT COMPUTES VALUES OF THE 2 DERIVATIVES.
C
C  X0 - THE INITIAL VALUE OF THE INDEPENDENT VARIABLE
C  T0 - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  TEND - AN ARRAY THAT RETURNS THE FINAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  H - THE INCREMENT TO T, THE STEP SIZE
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  TOL - TOLERANCE
C  TWRK - AN ARRAY USED TO HOLD INTERMEDIATE VALUES DURING THE
C           COMPUTATION. IT MUST BE DIMENSIONED OF SIZE 6 X 2.
C
C-------------------------------------------------------------------
C
      SUBROUTINE RKFSY1(DERIV1,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL)
      REAL*8 X0,T0(2),TEND(2),F(2),TWRK(6,2),H,K,K1,K2,TOL,DEL,
     &       ERROR,SUM,STOREH,XEND
      INTEGER I
C
C--------- INITIALIZE FOR INTERVAL [X0, X0+H] -----------------------
C
      XEND = X0+H
      STOREH = H
C
C-------- CHECK TO SEE IF FINISHED ---------------------------------
C
    1 IF(X0.GE.XEND) THEN
```

```
      H = STOREH
      RETURN
        ELSE
C
C-------- GET FIRST ESTIMATE OF THE DELTA X'S ---------------------
C
        CALL DERIV1(T0,F,DEL,K,K1,K2)
        DO 10 I = 1,2
           TWRK(1,I) = H*F(I)
           TEND(I) = T0(I)+TWRK(1,I)/4.0D0
   10 CONTINUE
C
C-------- GET SECOND ESTIMATE ---------------------------------------
C
        CALL DERIV1(TEND,F,DEL,K,K1,K2)
        DO 20 I = 1,2
           TWRK(2,I) = H*F(I)
           TEND(I) = T0(I)+(TWRK(1,I)*3.0D0+TWRK(2,I)*9.0D0)/32.0D0
   20 CONTINUE
C
C-------- GET THIRD ESTIMATE ----------------------------------------
C
        CALL DERIV1(TEND,F,DEL,K,K1,K2)
        DO 30 I = 1,2
           TWRK(3,I) = H*F(I)
           TEND(I) = T0(I)+(TWRK(1,I)*1932.0D0-TWRK(2,I)*7200.0D0
     &            + TWRK(3,I)*7296.0D0)/2197.0D0
   30 CONTINUE
C
C-------- GET FOURTH ESTIMATE ---------------------------------------
C
        CALL DERIV1(TEND,F,DEL,K,K1,K2)
        DO 40 I = 1,2
           TWRK(4,I) = H*F(I)
           TEND(I) = T0(I)+(439.0D0*TWRK(1,I)/216.0D0-8.0D0*TWRK(2,I)
     &      + 3680.0D0*TWRK(3,I)/513.0D0-845.0D0*TWRK(4,I)/4104.0D0)
   40 CONTINUE
C
C-------- GET FIFTH ESTIMATE ----------------------------------------
C
        CALL DERIV1(TEND,F,DEL,K,K1,K2)
        DO 50 I = 1,2
           TWRK(5,I) = H*F(I)
           TEND(I) = T0(I)-8.0D0*TWRK(1,I)/27.0+2.0D0*TWRK(2,I)
     &     - 3544.0D0*TWRK(3,I)/2565.0D0+1859.0D0*TWRK(4,I)/4104.0D0
     &     -11.0D0*TWRK(5,I)/40.0D0
   50 CONTINUE
C
C-------- GET SIXTH ESTIMATE ----------------------------------------
C
        CALL DERIV1(TEND,F,DEL,K,K1,K2)
        DO 60 I = 1,2
           TWRK(6,I) = H*F(I)
```

```
   60 CONTINUE
C
C-------- ESTIMATE THE ERROR BY COMPUTING THE DIFFERENCE BETWEEN
C         THE FOURTH AND FIFTH ORDER EQUATIONS
C
      SUM = 0.0D0
      ERROR = 0.0D0
      DO 70 I = 1,2
       SUM = DABS(TWRK(1,I)/360.0D0-128.0D0*TWRK(3,I)/4275.0D0
     &      - 2197.0D0*TWRK(4,I)/75240.0D0+TWRK(5,I)/50.0D0
     &      + 2.0D0*TWRK(6,I)/55.0D0)
       ERROR = DMAX1(ERROR,SUM)
   70 CONTINUE
C
C-------- IF ERROR IS LESS THAN TOLERANCE, COMPUTE X AT THE ---------
C         END OF THE INTERVAL FROM A WEIGHTED AVERAGE OF THE SIX
C         ESTIMATES AND RETURN
C
      IF(ERROR.LT.TOL) THEN
    DO 80 I = 1,2
         TEND(I) = T0(I)+16.0D0*TWRK(1,I)/135.0D0
     &         + 6656.0D0*TWRK(3,I)/12825.0D0
     &         + 28561.0D0*TWRK(4,I)/56430.0-9.0D0*TWRK(5,I)/50.0D0
     &         + 2.0D0*TWRK(6,I)/55.0D0
         T0(I) = TEND(I)
   80   CONTINUE
    X0 = X0+H
      ENDIF
C
C-------- IF ERROR IS GREATER THAN TOLERANCE, DECREASE STEP & -------
C         GO AGAIN
C
      IF(ERROR.GT.TOL) THEN
    H = H/2.0D0
      ENDIF
C
C-------- IF ERROR IS SIGNIFICANTLY LESS THAN TOLERANCE, RELAX ------
C         STEP
C
      IF(ERROR .LT. H*TOL/10.0D0) THEN
    H = H*2.0D0
      ENDIF
C
C-------- IF OVERSHOOT END, REDUCE STEP -----------------------------
C
      IF(X0 + H .GT. XEND) THEN
    H = XEND-X0
      ENDIF
C
      ENDIF
C
C
      GO TO 1
```

```
      END
C
C2E****** RTAN: DERIV1(T,F,DEL,K,K1,K2) ******************************
C
C  PARAMETERS:
C
C  T - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-------------------------------------------------------------------
C
      SUBROUTINE DERIV1(T,F,DEL,K,K1,K2)
      REAL*8 T(2),F(2),K,K1,K2,DEL
      F(1) = T(2)
      F(2) = K1*T(1)**4.0D0/(K*DEL) - K2/(K*DEL)
      RETURN
      END
C
C2F ***** RTAN: MDLIN1(FCN1,X1,X2,XR,TB,TE,DEL,K,K1,K2,HT) **********
                         ',/,,TB,TE,DEL,*******
C
C  PARAMETERS:
C
C  FCN1 - FUNCTION THAT COMPUTES VALUES FOR F. MUST BE DECLARED
C         EXTERNAL IN CALLING PROGRAM
C  X1,X2 - INITIAL VALUES OF X. F(X) MUST CHANGE SIGNS AT THESE
C          POINTS
C  XR - RETURNS THE ROOT TO THE MAIN PROGRAM
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  HT - FIN HEIGHT
C
C--------------------------------------------------------------------
C
      SUBROUTINE MDLIN1(FCN1,X1,X2,XR,TB,TE,DEL,K,K1,K2,HT)
      REAL*8 XERR,FSAVE,F1,F2,FR,X1,X2,XR,XTOL,FTOL,TB,TE,DEL,
     &       K,K1,K2,FCN1,HT
      INTEGER I,J,PASS
      CHARACTER*2 ANS
      DATA XTOL,FTOL/0.0001D0,0.00001D0/
C
C-------- INITIALIZE VALUES ------------------------------------------
C
      PRINT 200
      X1SAV = X1
```

```
      X2SAV = X2
      F1 = FCN1(X1,TB,TE,HT,DEL,K,K1,K2)
      F2 = FCN1(X2,TB,TE,HT,DEL,K,K1,K2)
      F1SAV = F1
      F2SAV = F2
      FSAVE = F1
C
C-------- INITIALIZE COUNTERS ------------------------------------------
C
      I = 0
      J = 1
      PASS = 0
C
C-------- LIMIT SEARCH TO INCREMENTS OF 50 ITERATIONS -----------------
C
    5 I = I + 1
      IF(I.GT.J*50) GOTO 10
      XR = X2-F2*(X2-X1)/(F2-F1)
      FR = FCN1(XR,TB,TE,HT,DEL,K,K1,K2)
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,I,XR,FR
      IF(PASS.GE.1) PASS = PASS + 1
C
C-------- CHECK STOPPING CRITERIA -----------------------------------
C
      XERR = DABS(X1-X2)/2.0D0
      IF(XERR.LE.XTOL.OR.DABS(FR).LE.FTOL) RETURN
C
C-------- FIND NEW POINT --------------------------------------------
C
      IF(FR*F1.GE.0.0D0) THEN
    X1 = XR
    F1 = FR
    IF(FR*FSAVE.GT.0.0D0) F2 = F2/2.0D0
        FSAVE = FR
      ELSE
        X2 = XR
        F2 = FR
      IF(FR*FSAVE.GT.0.0D0) F1 = F1/2.0D0
        FSAVE = FR
      ENDIF
      GOTO 5
C
C-------- FAIL TO FIND ROOT, CONTINUE SEARCH ? ---------------------
C
   10 PRINT 400,J*50
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
```

```
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- FAIL TO FIND ROOT, LOOK AT FUNTION VALUES ? ----------------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
    GOTO 15
      ENDIF
      IF(ANS.EQ.'N') STOP
      X1 = X1SAV
      X2 = X2SAV
      F1 = F1SAV
      F2 = F2SAV
      I = 0
      J = 1
      PASS = 1
      GOTO 5
C
C   ---------------------------------------------------------------------
C
  100 FORMAT('+', 'ITERATION : ',I3,'                              ')
  200 FORMAT(/,' COMMENCING SEARCH FOR ROOT IN INTERVAL....         ',/)
  300 FORMAT(' AT ITERATION',I3,3X,' X = ',D15.6,3X,' F(X) = ',D15.6)
  400 FORMAT(/,' FAILED TO FIND ROOT AFTER ',I3,' ITERATIONS.'       ,
     & ' CONTINUE SEARCH (Y OR N) ?'                                 )
  500 FORMAT(A2)
  600 FORMAT(/' ***************** INCORRECT ANSWER ******************')
  700 FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N )?   ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING.    ',/,
     & ' IF F(XR) IS OSCILLATING ABOUT A CERTAIN VALUE OR ITS       ',/,
     & ' ABSOLUTE VALUE IS INCREASING THEN THERE IS LITTLE          ',/,
     & ' PROBABILITY THAT A ROOT WILL BE FOUND.                      ')
      END
C
C3A****** RTSY: HEADER2 **********************************************
C
C  SUBROUTINE RTSY(TB,TE,L,HT,DEL,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
C  FIN SHAPE:  RECTANGLE
C
C  PROBLEM TYPE:  SYNTHESIS
C
C  INPUT:  TB,L,DEL,K,ABS,EMIS,E,Q
C
C  OUTPUT: HT,TE,QI,EFF
C
C-------------------------------------------------------------------
C
C  PARAMETERS:
```

```
C
C  TB - TEMPERATURE AT THE BASE
C  TE - TEMPERATURE AT THE TIP
C  HT - HEIGHT OF THE FIN
C  L - LENGTH OF THE FIN
C  DEL - THICKNESS OF THE FIN
C  K - CONDUCTIVITY OF THE FIN
C  ABS - ABSORPTIVITY FOR FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  QI - IDEAL HEAT DISSIPATED BY THE FIN
C  Q - REAL HEAT DISSIPATED BY THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  FLAG - 0 = CONVERGENCE, 1 = DIVERGENCE
C
C-----------------------------------------------------------------------
C
C  FUNCTIONS:
C
C  FCN2(X,TB,TE,L,DEL,K,K1,K2,Q):
C
C   FUNCTION WHOSE ROOT IS FOUND (BASE DERIV & TIP TEMP)
C
C-----------------------------------------------------------------------
C
C  SUBROUTINES:
C
C  RKFSY2(DERIV2,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL):
C
C   SOLVES SECOND ORDER NONLINEAR DE BY RUNGE-KUTTE-FELHBERG METHOD
C
C  DERIV2(T,F,DEL,K,K1,K2):
C
C   COMPUTES DERIVATIVES FOR RKFSY2
C
C  MDLIN2(FCN2,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,L,DEL,K,K1,K2,Q):
C
C   FINDS THE ROOT OF FCN2 BY THE METHOD OF MODIFIED LINEAR
C   INTERPOLATION
C
C3B ***** RTSY: MAIN2 ***************************************************
C
      SUBROUTINE RTSY(TB,TE,L,HT,DEL,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
      REAL*8 TB,TE,L,HT,DEL,K,ABS,EMIS,E,QI,Q,EFF,X1,X2,XR,
     &       SB,K1,K2,FCN2,F1,F2
      INTEGER I,J,PASS,FLAG
      CHARACTER*2 ANS
      EXTERNAL FCN2
C
C-------- DEFINE CONSTANTS ---------------------------------------------
C
      SB = 5.67051D-12
      K1 = 2.0D0*SB*EMIS
```

```
      K2 = E*ABS
C
C-------- SCALE INPUTS ---------------------------------------------
C
      DEL = DEL*100.0D0
      L = L*100.0D0
      K = K*0.01D0
C
C-------- START COUNTERS -------------------------------------------
C
      PASS = 0
      I = 0
      J = 1
C
C-------- START SEARCH WITH INTERVAL [0.1,0.2] ---------------------
C
      PRINT 200
      X1 = 0.1D0
      X2 = 0.2D0
      F1 = FCN2(X1,TB,TE,L,DEL,K,K1,K2,Q)
    5 I = I + 1
C
C-------- LIMIT SEARCH TO INCREMENTS OF 100 INTERVALS --------------
C
      IF(I.GT.J*100) GOTO 10
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,X1,F1,X2,F2
      F2 = FCN2(X2,TB,TE,L,DEL,K,K1,K2,Q)
C
C-------- CHECK FOR FUNCTION VALUES OPPOSITE IN SIGN ---------------
C
      IF(F1*F2.GE.0.0D0) THEN
        X1 = X2
        F1 = F2
        IF(X2.LT.1.0D0) X2 = X2 + 0.1D0
        IF(X2.GE.1.0D0) X2 = X2 + 1.0D0
        IF(PASS.GE.1) PASS = PASS + 1
        GOTO 5
      ENDIF
C
C-------- IF INTERVAL FOUND, GO FIND ROOT --------------------------
C
      PRINT 201
      GOTO 20
C
C-------- IF NO INTERVAL FOUND, CONTINUE SEARCH ? ------------------
C
   10 PRINT 400,J*100
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
```

```
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- IF UNABLE TO FIND INTERVAL, LOOK AT FUNCTION VALUES ? -------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 15
      ENDIF
      IF(ANS.EQ.'N') THEN
    FLAG = 1
    RETURN
      ENDIF
      PASS = 1
      I = 0
      J = 1
      X1 = 0.1D0
      X2 = 0.2D0
      F1 = FCN2(X1,TB,TE,L,DEL,K,K1,K2,Q)
      GOTO 5
C
C-------- GET ROOT IN INTERVAL: FIN HEIGHT AND TEMP AT TIP ----------
C
   20 CALL MDLIN2(FCN2,X1,X2,XR,TB,TE,L,DEL,K,K1,K2,Q)
      HT = XR
C
C-------- CALCULATE IDEAL HEAT DISSIPATED --------------------------
C
      QI = 2.0D0*SB*EMIS*HT*L*TB**4.0D0
C
C-------- CALCULATE FIN EFFICIENCY ---------------------------------
C
      EFF = Q/QI
C
C-------- SCALE OUTPUTS --------------------------------------------
C
      HT = HT*0.01D0
      DEL = DEL*0.01D0
      L = L*0.01D0
      K = K*100.0D0
C
C-------- FORMAT ---------------------------------------------------
C
100   FORMAT('+','LOOKING IN INTERVAL: ',I3,'                          ')
200   FORMAT(/,' COMMENCING SEARCH FOR INTERVAL TO BRACKET ROOT.... ',/)
201   FORMAT(/,' INTERVAL FOUND, WILL NOW LOOK FOR ROOT ....        ',/)
300   FORMAT(' XL =',2X,D10.4,2X,'F(XL) =',2X,D10.4,2X,
     &       'XR =',2X,D10.4,2X,'F(XR) =',2X,D10.4)
```

```
400   FORMAT(/,' NO ROOT FOUND IN ',I3,' INTERVALS.  CONTINUE SEARCH'  ,
     & ' (Y OR N) ?'                                                   )
500   FORMAT(A2)
600   FORMAT(/,' **************** INCORRECT ENTRY ! ***************** ')
700   FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N)?    ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING. IF ',/,
     & ' F(XR) IS OF THE SAME SIGN AS F(XL) AND ITS ABSOLUTE VALUE  ',/,
     & ' IS INCREASING, THERE IS LITTLE PROBABILITY OF FINDING AN   ',/,
     & ' INTERVAL.  THE SAME GOES FOR VALUES OF F(XR) THAT OSCILLATE',/,
     & ' AROUND A CERTAIN VALUE.  FOR F(XR) TO BECOME OPPOSITE IN   ',/,
     & ' SIGN TO F(XR), IT MUST PASS THROUGH ZERO.                  ')
      RETURN
      END
C
C3C****** RTSY: FCN2(X,TB,TE,L,DEL,K,K1,K2,Q) **********************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE (BASE DERIV)
C  TB - BASE TEMP
C  TE - TIP TEMP
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = ABS*E
C  Q - HEAT INPUT THRU BASE
C
C------------------------------------------------------------------
C
      REAL*8 FUNCTION FCN2(X,TB,TE,L,DEL,K,K1,K2,Q)
      REAL*8 X,X0,XFINAL,T0(2),TEND(2),F(2),H,TB,TE,DEL,
     &       K,K1,K2,L,Q,TSAVE,TOL
      EXTERNAL DERIV2
C
C-------- INITIALIZE LEFT END BC'S: KNOW FCT VALUE & DERIV ----------
C
      T0(1)= TB
      T0(2)= -Q/(K*DEL*L)
C
C-------- INITIALIZE RIGHT END BC'S TO 0 ----------------------------
C
      TEND(1)=0.0D0
      TEND(2)=0.0D0
C
C-------- DEFINE PARAMETERS ------------------------------------------
C
      IF (X.LT.1.0) H = 0.05D0
      IF (X.GE.1.0) H = 0.1D0
      TOL = 0.0001D0
      X0 = 0.0D0
      XFINAL = X
      TSAVE = 1000.0D0
10    IF(X0.LT.XFINAL+0.0001D0) THEN
```

```
      CALL RKFSY2(DERIV2,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL)
C
C-------- INCREASING T MEANS DIVERGENCE ----------------------------
C
      IF(TEND(1).GE.TSAVE) GO TO 20
      T0(1) = TEND(1)
      TSAVE = TEND(1)
      T0(2) = TEND(2)
      GOTO 10
        ENDIF
20      CONTINUE
        FCN2 = TEND(2)
        TE = TEND(1)
        RETURN
        END
C
C3D ***** RTSY: RKFSY2(DERIV2,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL)
C
C  PARAMTERS:
C
C  RKFSY2 - SUBROUTINE THAT SOLVES A SYSTEM OF 2 FIRST ORDER
C           DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-FEHLBERG
C           METHOD. THE EQUATIONS ARE OF THE FORM:
C
C           DT/DX = Y = F1(X,T)
C           DY/DX = F2(X,T,Y)
C
C  DERIV2 - A SUBROUTINE THAT COMPUTES VALUES OF THE 2 DERIVATIVES.
C
C  X0 - THE INITIAL VALUE OF INDEPENDENT VARIABLE
C  T0 - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  TEND - AN ARRAY THAT RETURNS THE FINAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  H - THE INCREMENT TO T, THE STEP SIZE
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  TOL - TOLERANCE
C  TWRK - AN ARRAY USED TO HOLD INTERMEDIATE VALUES DURING THE
C           COMPUTATION. IT MUST BE DIMENSIONED OF SIZE 6 X 2.
C
C-------------------------------------------------------------------
C
        SUBROUTINE RKFSY2(DERIV2,X0,T0,TEND,F,H,DEL,K,K1,K2,TOL)
        REAL*8 X0,T0(2),TEND(2),F(2),TWRK(6,2),H,K,K1,K2,TOL,DEL,
     &        ERROR,SUM,STOREH,XEND
        INTEGER I
C
C--------- INITIALIZE FOR INTERVAL [X0, X0+H] ----------------------
C
        XEND = X0+H
        STOREH = H
```

```
C
C-------- CHECK TO SEE IF WE ARE FINISHED -------------------------
C
    1 IF(X0.GE.XEND) THEN
    H = STOREH
    RETURN
      ELSE
C
C-------- GET FIRST ESTIMATE OF THE DELTA X'S ---------------------
C
      CALL DERIV2(T0,F,DEL,K,K1,K2)
      DO 10 I = 1,2
         TWRK(1,I) = H*F(I)
         TEND(I) = T0(I)+TWRK(1,I)/4.0D0
   10 CONTINUE
C
C-------- GET SECOND ESTIMATE ---------------------------------------
C
      CALL DERIV2(TEND,F,DEL,K,K1,K2)
      DO 20 I = 1,2
         TWRK(2,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*3.0D0+TWRK(2,I)*9.0D0)/32.0D0
   20 CONTINUE
C
C-------- GET THIRD ESTIMATE ----------------------------------------
C
      CALL DERIV2(TEND,F,DEL,K,K1,K2)
      DO 30 I = 1,2
         TWRK(3,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*1932.0D0-TWRK(2,I)*7200.0D0
     &            + TWRK(3,I)*7296.0D0)/2197.0D0
   30 CONTINUE
C
C-------- GET FOURTH ESTIMATE ---------------------------------------
C
      CALL DERIV2(TEND,F,DEL,K,K1,K2)
      DO 40 I = 1,2
         TWRK(4,I) = H*F(I)
         TEND(I) = T0(I)+(439.0D0*TWRK(1,I)/216.0D0-8.0D0*TWRK(2,I)
     &      + 3680.0D0*TWRK(3,I)/513.0D0-845.0D0*TWRK(4,I)/4104.0D0)
   40 CONTINUE
C
C-------- GET FIFTH ESTIMATE ----------------------------------------
C
      CALL DERIV2(TEND,F,DEL,K,K1,K2)
      DO 50 I = 1,2
         TWRK(5,I) = H*F(I)
         TEND(I) = T0(I)-8.0D0*TWRK(1,I)/27.0+2.0D0*TWRK(2,I)
     &     - 3544.0D0*TWRK(3,I)/2565.0D0+1859.0D0*TWRK(4,I)/4104.0D0
     &     - 11.0D0*TWRK(5,I)/40.0D0
50    CONTINUE
C
C-------- GET SIXTH ESTIMATE ----------------------------------------
```

```
C
      CALL DERIV2(TEND,F,DEL,K,K1,K2)
      DO 60 I = 1,2
         TWRK(6,I) = H*F(I)
60    CONTINUE
C
C-------- ESTIMATE THE ERROR BY COMPUTING THE DIFFERENCE BETWEEN
C           THE FOURTH AND FIFTH ORDER EQUATIONS
C
      SUM = 0.0D0
      ERROR = 0.0D0
      DO 70 I = 1,2
       SUM = DABS(TWRK(1,I)/360.0D0-128.0D0*TWRK(3,I)/4275.0D0
     &      - 2197.0D0*TWRK(4,I)/75240.0D0+TWRK(5,I)/50.0D0
     &      + 2.0D0*TWRK(6,I)/55.0D0)
       ERROR = DMAX1(ERROR,SUM)
70    CONTINUE
C
C-------- IF ERROR IS LESS THAN TOLERANCE, COMPUTE X AT
C           END OF THE INTERVAL FROM A WEIGHTED AVERAGE OF THE SIX
C           ESTIMATES & THEN RETURN
C
      IF(ERROR.LT.TOL) THEN
    DO 80 I = 1,2
       TEND(I) = T0(I)+16.0D0*TWRK(1,I)/135.0D0
     &      + 6656.0D0*TWRK(3,I)/12825.0D0
     &      + 28561.0D0*TWRK(4,I)/56430.0-9.0D0*TWRK(5,I)/50.0D0
     &      + 2.0D0*TWRK(6,I)/55.0D0
       T0(I) = TEND(I)
80       CONTINUE
         X0 = X0+H
      ENDIF
C
C-------- IF ERROR IS GREATER THAN TOLERANCE, DECREASE STEP & -------
C           GO AGAIN
C
      IF(ERROR.GT.TOL) THEN
    H = H/2.0D0
      ENDIF
C
C-------- IF ERROR IS SIGNIFICANTLY LESS THAN TOLERANCE, RELAX STEP --
C
      IF(ERROR.LT.H*TOL/10.0D0) THEN
    H = H*2.0D0
      ENDIF
C
C-------- IF OVERSHOOT, REDUCE STEP ---------------------------------
C
      IF(X0 + H.GT.XEND) THEN
    H = XEND - X0
      ENDIF
C
      ENDIF
```

```
C
C
      GO TO 1
      END
C
C3E****** RTSY: DERIV2(T,F,DEL,K,K1,K2) *****************************
C
C  PARAMETERS:
C
C  T - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-------------------------------------------------------------------
C
      SUBROUTINE DERIV2(T,F,DEL,K,K1,K2)
      REAL*8 T(2),F(2),K,K1,K2,DEL
      F(1) = T(2)
      F(2) = K1*T(1)**4.0D0/(K*DEL) - K2/(K*DEL)
      RETURN
      END
C
C3F ***** RTSY: MDLIN2(FCN2,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,L,DEL,
C                K,K1,K2,Q)
C
C  PARAMETERS:
C
C  FCN2 - FUNCTION THAT COMPUTES VALUES FOR F. MUST BE DECLARED
C         EXTERNAL IN CALLING PROGRAM
C  X1,X2 - INITIAL VALUES OF X. F(X) MUST CHANGE SIGNS AT THESE
C          POINTS
C  XR - RETURNS THE ROOT TO THE MAIN PROGRAM
C  XTOL - TOLERANCE FOR X
C  FTOL - TOLERANCE FOR F
C  NLIM - LIMIT TO NUMBER OF ITERATIONS
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  L - FIN LENGTH
C  DEL - FIN THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  Q - HEAT INPUT THRU BASE
C
C--------------------------------------------------------------------
C
      SUBROUTINE MDLIN2(FCN2,X1,X2,XR,TB,TE,L,DEL,K,K1,K2,Q)
      REAL*8 XERR,FSAVE,F1,F2,FR,X1,X2,XR,XTOL,FTOL,TB,TE,L,DEL,
     &       K,K1,K2,Q,FCN2,X1SAV,X2SAV,F1SAV,F2SAV
      INTEGER I,J,PASS
```

```
      CHARACTER*2 ANS
      DATA XTOL,FTOL/0.0001D0,0.00001D0/
C
C-------- INITIALIZE VALUES -------------------------------------------
C
      PRINT 200
      X1SAV = X1
      X2SAV = X2
      F1 = FCN2(X1,TB,TE,L,DEL,K,K1,K2,Q)
      F2 = FCN2(X2,TB,TE,L,DEL,K,K1,K2,Q)
      F1SAV = F1
      F2SAV = F2
      FSAVE = F1
C
C-------- INITIALIZE COUNTER ------------------------------------------
C
      I = 0
      J = 1
      PASS = 0
C
C-------- LIMIT SEARCH TO INCREMENTS OF 50 ITERATIONS -----------------
C
    5 I = I + 1
      IF(I.GT.J*50) GOTO 10
      XR = X2-F2*(X2-X1)/(F2-F1)
      FR = FCN2(XR,TB,TE,L,DEL,K,K1,K2,Q)
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,I,XR,FR
      IF(PASS.GE.1) PASS = PASS + 1
C
C-------- CHECK STOPPING CRITERIA ------------------------------------
C
      XERR = DABS(X1-X2)/2.0D0
      IF(XERR.LE.XTOL.OR.DABS(FR).LE.FTOL) RETURN
C
C-------- FIND NEW POINT ---------------------------------------------
C
      IF(FR*F1.GE.0.0D0) THEN
    X1 = XR
    F1 = FR
    IF(FR*FSAVE.GT.0.0D0) F2 = F2/2.0D0
    FSAVE = FR
      ELSE
    X2 = XR
    F2 = FR
    IF(FR*FSAVE.GT.0.0D0) F1 = F1/2.0D0
    FSAVE = FR
      ENDIF
      GOTO 5
C
C-------- FAIL TO FIND ROOT, CONTINUE SEARCH ? ---------------------
C
```

```
   10 PRINT 400,J*50
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- FAIL TO FIND ROOT, LOOK AT FUNTION VALUES ? ----------------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
    GOTO 15
      ENDIF
      IF(ANS.EQ.'N') STOP
      X1 = X1SAV
      X2 = X2SAV
      F1 = F1SAV
      F2 = F2SAV
      I = 0
      J = 1
      PASS = 1
      GOTO 5
C
C  ---------------------------------------------------------------------
C
  100 FORMAT('+', 'ITERATION : ',I3,'                              ')
  200 FORMAT(/,' COMMENCING SEARCH FOR ROOT IN INTERVAL....         ',/)
  300 FORMAT(' AT ITERATION',I3,3X,' X = ',D15.6,3X,' F(X) = ',D15.6)
  400 FORMAT(/,' FAILED TO FIND ROOT AFTER ',I3,' ITERATIONS.'         ,
     & ' CONTINUE SEARCH (Y OR N) ?'                                   )
  500 FORMAT(A2)
  600 FORMAT(/' ***************** INCORRECT ANSWER ******************')
  700 FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N )?   ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING.    ',/,
     & ' IF F(XR) IS OSCILLATING ABOUT A CERTAIN VALUE OR ITS       ',/,
     & ' ABSOLUTE VALUE IS INCREASING THEN THERE IS LITTLE          ',/,
     & ' PROBABILITY THAT A ROOT WILL BE FOUND.                      ')
      END
C
C4A ****** TPAN: HEADER3************************************************
C
C  SUBROUTINE TPAN(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
C  FIN SHAPE:  TRAPEZOID
C
C  PROBLEM TYPE:  ANALYSIS
C
```

```
C  INPUT:  TB,L,HT,DEL0,DELE,K,ABS,EMIS,E
C
C  OUTPUT: Q,TE,QI,EFF
C
C-----------------------------------------------------------------------
C
C  PARAMETERS:
C
C  TB - TEMPERATURE AT THE BASE OF THE FIN
C  TE - TEMPERATURE AT THE TIP OF THE FIN
C  HT - HEIGHT OF THE FIN
C  L - LENGTH OF THE FIN
C  DEL0 - THICKNESS AT FIN BASE
C  DELE - THICKNESS AT FIN TIP
C  K - CONDUCTIVITY OF THE FIN
C  ABS - ABSORPTIVITY OF THE FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  QI - IDEAL HEAT DISSIPATED BY THE FIN
C  Q - REAL HEAT DISSIPATED BY THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  FLAG - 0 = CONVERGENCE, 1 = DIVERGENCE
C
C-----------------------------------------------------------------------
C
C  FUNCTIONS:
C
C  FCN3(X,TB,TE,HT,DEL0,DELE,K,K1,K2):
C
C   FUNCTION WHOSE ROOT IS FOUND (BASE DERIV & TIP TEMP)
C
C-----------------------------------------------------------------------
C
C  SUBROUTINES:
C
C  RKFSY3(DERIV3,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL):
C
C   SOLVES A SECOND ORDER NONLINEAR DE BY RUNGE-KUTTE-FEHLBERG METHOD
C
C  DERIV3(X,T,F,HT,DEL0,DELE,K,K1,K2):
C
C   COMPUTES DERIVATIVES FOR RKFSY3
C
C  MDLIN3(FCN3,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,HT,DEL0,DELE,K,K1,K2):
C
C   FINDS THE ROOT OF EQUATION FCN3 BY THE METHOD OF MODIFIED
C   LINEAR INTERPOLATION
C
C4B ***** TPAN: MAIN3**************************************************
C
```

```
      SUBROUTINE TPAN(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
      REAL*8 TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,X1,X2,XR,
     &       SB,TBDER,K1,K2,F1,F2,FCN3
      INTEGER I,J,PASS,FLAG
      CHARACTER*2 ANS
      EXTERNAL FCN3
C
C-------- DEFINE CONSTANTS -------------------------------------------
C
      SB = 5.67051D-12
      K1 = 2.0D0*SB*EMIS
      K2 = E*ABS
C
C-------- SCALE INPUTS -----------------------------------------------
C
      HT = HT*100.0D0
      DEL0 = DEL0*100.0D0
      DELE = DELE*100.0D0
      L = L*100.0D0
      K = K*0.01D0
C
C-------- START COUNTERS ---------------------------------------------
C
      PASS = 0
      I = 0
      J = 1
C
C-------- START SEARCH WITH INTERVAL [-1,0] --------------------------
C
      PRINT 200
      X1 = 0.0D0
      X2 = -1.0D0
      F1 = FCN3(X1,TB,TE,HT,DEL0,DELE,K,K1,K2)
    5 I = I + 1
C
C-------- LIMIT SEARCH TO INCREMENTS OF 100 INTERVALS ---------------
C
      IF(I.GT.J*100) GOTO 10
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,X1,F1,X2,F2
      F2 = FCN3(X2,TB,TE,HT,DEL0,DELE,K,K1,K2)
C
C-------- CHECK FOR FUNCTION VALUES OPPOSITE IN SIGN ----------------
C
      IF(F1*F2.GE.0.0D0) THEN
        X1 = X2
        F1 = F2
        X2 = X2 - 1.0D0
        IF(PASS.GE.1) PASS = PASS + 1
```

```
        GOTO 5
      ENDIF
C
C-------- IF INTERVAL FOUND, GO FIND ROOT ---------------------------
C
      PRINT 201
      GOTO 20
C
C-------- IF NO INTERVAL FOUND, CONTINUE SEARCH ? ------------------
C
   10 PRINT 400,J*100
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- IF UNABLE TO FIND INTERVAL, LOOK AT FUNCTION VALUES ? -------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 15
      ENDIF
      IF(ANS.EQ.'N') THEN
    FLAG = 1
    RETURN
      ENDIF
      PASS = 1
      I = 0
      J = 1
      X1 = 0.0D0
      X2 = -1.0D0
      F1 = FCN3(X1,TB,TE,HT,DEL0,DELE,K,K1,K2)
      GOTO 5
C
C-------- GET ROOT IN INTERVAL:  DERIV AT BASE & TEMP AT TIP --------
C
   20 CALL MDLIN3(FCN3,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2)
      TBDER = XR
C
C-------- CALCULATE IDEAL HEAT DISSIPATED ---------------------------
C
      QI = 2.0D0*SB*EMIS*HT*L*TB**4.0D0
C
```

```
C--------- CALCULATE REAL*8 HEAT DISSIPATED ---------------------------
C
      Q = -K*DEL0*L*TBDER
C
C-------- CALCULATE FIN EFFICIENCY ---------------------------------
C
      EFF = Q/QI
C
C-------- SCALE OUTPUTS ----------------------------------------------
C
      HT = HT*0.01D0
      DEL0 = DEL0*0.01D0
      DELE = DELE*0.01D0
      L = L*0.01D0
      K = K*100.0D0
      RETURN
C
C-------- FORMAT ------------------------------------------------------
C
100   FORMAT('+','LOOKING IN INTERVAL: ',I3,'                         ')
200   FORMAT(/,' COMMENCING SEARCH FOR INTERVAL TO BRACKET ROOT.... ',/)
201   FORMAT(/,' INTERVAL FOUND, WILL NOW LOOK FOR ROOT ....        ',/)
300   FORMAT(' XL =',2X,D10.4,2X,'F(XL) =',2X,D10.4,2X,
     &        'XR =',2X,D10.4,2X,'F(XR) =',2X,D10.4)
400   FORMAT(/,' NO ROOT FOUND IN ',I3,' INTERVALS.  CONTINUE SEARCH'  ,
     & ' (Y OR N) ?'                                                   )
500   FORMAT(A2)
600   FORMAT(/,' **************** INCORRECT ENTRY ! ***************** ')
700   FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N)?     ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING. IF ',/,
     & ' F(XR) IS OF THE SAME SIGN AS F(XL) AND ITS ABSOLUTE VALUE  ',/,
     & ' IS INCREASING, THERE IS LITTLE PROBABILITY OF FINDING AN   ',/,
     & ' INTERVAL.  THE SAME GOES FOR VALUES OF F(XR) THAT OSCILLATE',/,
     & ' AROUND A CERTAIN VALUE.  FOR F(XR) TO BECOME OPPOSITE IN   ',/,
     & ' SIGN TO F(XR), IT MUST PASS THROUGH ZERO.                  ')
      END
C
C4C****** TPAN: FCN3(X,TB,TE,HT,DEL0,DELE,K,K1,K2) ******************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE (BASE DERIV)
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = ABS*E
```

```
C
C-------------------------------------------------------------------
C
      REAL*8 FUNCTION FCN3(X,TB,TE,HT,DEL0,DELE,K,K1,K2)
      REAL*8 X,X0,XFINAL,T0(2),TEND(2),F(2),H,TB,TE,HT,DEL0,DELE,
     &       K,K1,K2,TSAVE,TOL
      EXTERNAL DERIV3
C
C-------- INITIALIZE LEFT END POINT BC'S: KNOW FCT VALUE, GUESS
C         DERIVATIVE
C
      T0(1)= TB
      T0(2)= X
C
C-------- INITIALIZE RIGHT END POINT BC'S TO 0 ----------------------
C
      TEND(1)=0.0D0
      TEND(2)=0.0D0
C
C-------- DEFINE PARAMETERS -----------------------------------------
C
      H= 0.1D0
      TOL = 0.0001D0
      X0 = 0.0D0
      XFINAL = HT
      TSAVE = 1000.0D0
10    IF(X0.LT.XFINAL+0.0001D0) THEN
    CALL RKFSY3(DERIV3,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
C
C-------- INCREASING T MEANS DIVERGENCE ----------------------------
C
    IF (TEND(1).GE.TSAVE) GOTO 20
    T0(1) = TEND(1)
    TSAVE = TEND(1)
    T0(2) = TEND(2)
    GO TO 10
      ENDIF
20    CONTINUE
      FCN3 = TEND(2)
      TE = TEND(1)
      RETURN
      END
C
C4D ***** TPAN: RKFSY3(DERIV3,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
C
C  PARAMTERS:
C
C  RKFSY3 - SUBROUTINE THAT SOLVES A SYSTEM OF 2 FIRST ORDER
C           DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-FEHLBERG
C           METHOD. THE EQUATIONS ARE OF THE FORM:
```

```
C
C           DT/DX = Y = F1(X,T)
C           DY/DX = F2(X,T,Y)
C
C  DERIV3 - A SUBROUTINE THAT COMPUTES VALUES OF THE 2 DERIVATIVES.
C
C  X0 - THE INITIAL VALUE OF INDEPENDENT VARIABLE
C  H - THE INCREMENT TO T, THE STEP SIZE
C  T0 - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  TEND - AN ARRAY THAT RETURNS THE FINAL VALUES OF THE FUNCTIONS
C  H - STEP SIZE
C  HT - HEIGHT OF FIN
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  DEL0 - THICKNESS AT BASE
C  DELE - THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  TOL - TOLERANCE
C  TWRK - AN ARRAY USED TO HOLD INTERMEDIATE VALUES DURING THE
C           COMPUTATION. IT MUST BE DIMENSIONED OF SIZE 6 X 2.
C
C-------------------------------------------------------------------
C
      SUBROUTINE RKFSY3(DERIV3,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
      REAL*8 X0,T0(2),TEND(2),F(2),TWRK(6,2),H,HT,DEL0,DELE,K,K1,K2,
     &     TOL,ERROR,SUM,STOREH,XEND
      INTEGER I
C
C-------- INITIALIZE FOR INTERVAL [X0, X0+H] ------------------------
C
      XEND = X0+H
      STOREH = H
C
C-------- CHECK TO SEE IF WE ARE FINISHED ---------------------------
C
1     IF(X0.GE.XEND) THEN
    H = STOREH
    RETURN
      ELSE
C
C-------- GET FIRST ESTIMATE OF THE DELTA X'S ------------------------
C
      CALL DERIV3(X0,T0,F,HT,DEL0,DELE,K,K1,K2)
      DO 10 I = 1,2
         TWRK(1,I) = H*F(I)
         TEND(I) = T0(I)+TWRK(1,I)/4.0D0
   10 CONTINUE
C
C-------- GET SECOND ESTIMATE ---------------------------------------
```

```
C
      CALL DERIV3(X0+H/4.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 20 I = 1,2
         TWRK(2,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*3.0D0+TWRK(2,I)*9.0D0)/32.0D0
   20 CONTINUE
C
C-------- GET THIRD ESTIMATE ---------------------------------------
C
      CALL DERIV3(X0+3.0D0*H/8.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 30 I = 1,2
         TWRK(3,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*1932.0D0-TWRK(2,I)*7200.0D0
     &                + TWRK(3,I)*7296.0D0)/2197.0D0
   30 CONTINUE
C
C-------- GET FOURTH ESTIMATE --------------------------------------
C
      CALL DERIV3(X0+12.0*H/13.0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 40 I = 1,2
         TWRK(4,I) = H*F(I)
         TEND(I) = T0(I)+(439.0D0*TWRK(1,I)/216.0D0-8.0D0*TWRK(2,I)
     &     + 3680.0D0*TWRK(3,I)/513.0D0-845.0D0*TWRK(4,I)/4104.0D0)
   40 CONTINUE
C
C-------- GET FIFTH ESTIMATE ---------------------------------------
C
      CALL DERIV3(X0+H,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 50 I = 1,2
         TWRK(5,I) = H*F(I)
         TEND(I) = T0(I)-8.0D0*TWRK(1,I)/27.0+2.0D0*TWRK(2,I)
     &     - 3544.0D0*TWRK(3,I)/2565.0D0+1859.0D0*TWRK(4,I)/4104.0D0
     &     - 11.0D0*TWRK(5,I)/40.0D0
50    CONTINUE
C
C-------- GET SIXTH ESTIMATE ---------------------------------------
C
      CALL DERIV3(X0+H/2.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 60 I = 1,2
         TWRK(6,I) = H*F(I)
60    CONTINUE
C
C-------- ESTIMATE ERROR BY COMPUTING DIFFERENCE BETWEEN ------------
C        THE FOURTH AND FIFTH ORDER EQUATIONS
C
      SUM = 0.0D0
      ERROR = 0.0D0
      DO 70 I = 1,2
         SUM = DABS(TWRK(1,I)/360.0D0-128.0D0*TWRK(3,I)/4275.0D0
     &         - 2197.0D0*TWRK(4,I)/75240.0D0+TWRK(5,I)/50.0D0
```

```
     &        + 2.0D0*TWRK(6,I)/55.0D0)
          ERROR = DMAX1(ERROR,SUM)
70    CONTINUE
C
C-------- IF ERROR LESS THAN TOLERANCE, COMPUTE X AT THE END OF
C         INTERVAL FROM A WEIGHTED AVERAGE OF THE SIX ESTIMATES &
C         THEN RETURN
C

      IF(ERROR.LT.TOL) THEN
    DO 80 I = 1,2
       TEND(I) = T0(I)+16.0D0*TWRK(1,I)/135.0D0
     &       + 6656.0D0*TWRK(3,I)/12825.0D0
     &       + 28561.0D0*TWRK(4,I)/56430.0-9.0D0*TWRK(5,I)/50.0D0
     &       + 2.0D0*TWRK(6,I)/55.0D0
       T0(I) = TEND(I)
80  CONTINUE
    X0 = X0+H
      ENDIF
C
C-------- IF ERROR GREATER THAN TOLERANCE, THEN REDUCE STEP ---------
C
      IF(ERROR.GT.TOL) THEN
    H = H/2.0D0
      ENDIF
C
C-------- IF ERROR IS SIGNIFICANTLY LESS THAN TOLERANCE, RELAX STEP
C
      IF(ERROR.LT.H*TOL/10.0D0) THEN
    H = H*2.0D0
      ENDIF
C
C-------- IF OVERSHOOT, REDUCE STEP ---------------------------------
C
      IF(X0+H.GT.XEND) THEN
    H = XEND-X0
      ENDIF
C
      ENDIF
C
      GO TO 1
      END
C
C4E****** TPAN: DERIV3(X,T,F,HT,DEL0,DELE,K,K1,K2) ******************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE
C  T - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
```

```
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-----------------------------------------------------------------------
C
      SUBROUTINE DERIV3(X,T,F,HT,DEL0,DELE,K,K1,K2)
      REAL*8 X,T(2),F(2),HT,DEL0,DELE,K,K1,K2,DEL,DELP
      DEL(X) = DEL0+(DELE-DEL0)*X/HT
      DELP = (DELE-DEL0)/HT
      F(1) = T(2)
      F(2) = -(DELP*T(2))/DEL(X)+(K1*T(1)**4.0D0-K2)/(K*DEL(X))
      RETURN
      END
C
C4F ***** TPAN:MDLIN3(FCN3,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,HT,DEL0,
C                       DELE,K,K1,K2)
C
C  PARAMETERS:
C
C  FCN3 - FUNCTION THAT COMPUTES VALUES FOR F. MUST BE DECLARED
C         EXTERNAL IN CALLING PROGRAM
C  X1,X2 - INITIAL VALUES OF X. F(X) MUST CHANGE SIGNS AT THESE
C          POINTS
C  XR - RETURNS THE ROOT TO THE MAIN PROGRAM
C  XTOL - TOLERANCE FOR X
C  FTOL - TOLERANCE FOR F
C  NLIM - LIMIT TO NUMBER OF ITERATIONS
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - BASE THICKNESS
C  DELE - TIP THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-----------------------------------------------------------------------
C
      SUBROUTINE MDLIN3(FCN3,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2)
      REAL*8 XERR,FSAVE,F1,F2,FR,X1,X2,XR,XTOL,FTOL,TB,TE,HT,DEL0,
     &       DELE,K,K1,K2,FCN3,X1SAV,X2SAV,F1SAV,F2SAV
      INTEGER I,J,PASS
      CHARACTER*2 ANS
      DATA XTOL,FTOL/0.0001D0,0.00001D0/
C
C-------- INITIALIZE VALUES ------------------------------------------
C
```

```
      PRINT 200
      X1SAV = X1
      X2SAV = X2
      F1 = FCN3(X1,TB,TE,HT,DEL0,DELE,K,K1,K2)
      F2 = FCN3(X2,TB,TE,HT,DEL0,DELE,K,K1,K2)
      F1SAV = F1
      F2SAV = F2
      FSAVE = F1
C
C-------- INITIALIZE COUNTER ------------------------------------------
C
      I = 0
      J = 1
      PASS = 0
C
C-------- LIMIT SEARCH TO INCREMENTS OF 50 ITERATIONS -----------------
C
    5 I = I + 1
      IF(I.GT.J*50) GOTO 10
      XR = X2-F2*(X2-X1)/(F2-F1)
      FR = FCN3(XR,TB,TE,HT,DEL0,DELE,K,K1,K2)
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,I,XR,FR
      IF(PASS.GE.1) PASS = PASS + 1
C
C-------- CHECK STOPPING CRITERIA ------------------------------------
C
      XERR = DABS(X1-X2)/2.0D0
      IF(XERR.LE.XTOL.OR.DABS(FR).LE.FTOL) RETURN
C
C-------- FIND NEW POINT ---------------------------------------------
C
      IF(FR*F1.GE.0.0D0) THEN
    X1 = XR
    F1 = FR
    IF (FR*FSAVE.GT.0.0D0) F2 = F2/2.0D0
    FSAVE = FR
      ELSE
    X2 = XR
    F2 = FR
    IF(FR*FSAVE.GT.0.0D0) F1 = F1/2.0D0
    FSAVE = FR
      ENDIF
      GOTO 5
C
C-------- FAIL TO FIND ROOT, CONTINUE SEARCH ? ----------------------
C
   10 PRINT 400,J*50
      READ 500,ANS
```

```
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- FAIL TO FIND ROOT, LOOK AT FUNCTION VALUES ? ---------------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
    GOTO 15
      ENDIF
      IF(ANS.EQ.'N') STOP
      X1 = X1SAV
      X2 = X2SAV
      F1 = F1SAV
      F2 = F2SAV
      I = 0
      J = 1
      PASS = 1
      GOTO 5
C
C  --------------------------------------------------------------------
C
  100 FORMAT('+', 'ITERATION : ',I3,'                                ')
  200 FORMAT(/,' COMMENCING SEARCH FOR ROOT IN INTERVAL....          ',/)
  300 FORMAT(' AT ITERATION',I3,3X,' X = ',D15.6,3X,' F(X) = ',D15.6)
  400 FORMAT(/,' FAILED TO FIND ROOT AFTER ',I3,' ITERATIONS.'          ,
     & ' CONTINUE SEARCH (Y OR N) ?'                                   )
  500 FORMAT(A2)
  600 FORMAT(/' **************** INCORRECT ANSWER ******************')
  700 FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N )?   ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING.    ',/,
     & ' IF F(XR) IS OSCILLATING ABOUT A CERTAIN VALUE OR ITS       ',/,
     & ' ABSOLUTE VALUE IS INCREASING THEN THERE IS LITTLE          ',/,
     & ' PROBABILITY THAT A ROOT WILL BE FOUND.                      ')
      END
C
C5A ***** TPSY HEADER4 ***********************************************
C
C  SUBROUTINE TPSY(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
C  FIN SHAPE:  TRAPEZOID
C
C  PROBLEM TYPE:  SYNTHESIS
```

```
C
C  INPUT:  TB,L,DEL0,DELE,K,ABS,EMIS,E,Q
C
C  OUTPUT: TE,HT,QI,EFF
C
C-----------------------------------------------------------------------
C
C  PARAMETERS:
C
C  TB - TEMPERATURE AT THE BASE OF THE FIN
C  TE - TEMPERATURE AT THE TIP OF THE FIN
C  L - LENGTH OF THE FIN
C  HT - HEIGHT OF THE FIN
C  DEL0 - THICKNESS AT FIN BASE
C  DELE - THICKNESS AT FIN TIP
C  K - CONDUCTIVITY OF THE FIN
C  ABS - ABSORPTIVITY OF THE FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  QI - IDEAL HEAT DISSIPATED BY THE FIN
C  Q - REAL HEAT DISSIPATED BY THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  FLAG - 0 = CONVERGENCE, 1 = DIVERGENCE
C
C-----------------------------------------------------------------------
C
C  FUNCTIONS: FCN4(X,TB,TE,L,HT,DEL0,DELE,K,K1,K2,Q)
C
C   FUNCTION WHOSE ROOT IS FOUND (BASE DERIV & TIP TEMP)
C
C-----------------------------------------------------------------------
C
C  SUBROUTINES:
C
C  MDLIN4(FCN4,X1,X2,XR,XTOL,FTOL,NLIM,TB,HT,DEL,K,K1,K2):
C
C   FINDS THE ROOT OF AN EQUATION FCN4 BY THE METHOD OF MODIFIED
C   LINEAR INTERPOLATION
C
C  RKFSY4(DERIV4,XBEGIN,H,T0,TEND,F,HT,DEL0,DELE,K,K1,K2,TOL):
C
C    SOLVES SECOND ORDER NONLINEAR DE BY RUNGE-KUTTE-FEHLBERG METHOD
C
C  DERIV4(X,T,F,HT,DEL0,DELE,K,K1,K2):
C
C   COMPUTES DERIVATIVES FOR RKFSY4
C
C5B ***** TPSY: MAIN4 **************************************************
C
      SUBROUTINE TPSY(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
```

```
      REAL*8 TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,X1,X2,XR,
     &       SB,K1,K2,F1,F2,FCN4
      INTEGER I,J,PASS,FLAG
      CHARACTER*2 ANS
      EXTERNAL FCN4
C
C-------- DEFINE CONSTANTS -----------------------------------------
C
      SB = 5.67051D-12
      K1 = 2.0D0*SB*EMIS
      K2 = E*ABS
C
C-------- SCALE INPUTS ---------------------------------------------
C
      DEL0 = DEL0*100.0D0
      DELE = DELE*100.0D0
      L = L*100.0D0
      K = K*0.01D0
C
C-------- START COUNTERS -------------------------------------------
C
      PASS = 0
      I = 0
      J = 1
C
C-------- START SEARCH WITH INTERVAL [0.1,0.2] ---------------------
C
      PRINT 200
      X1 = 0.1D0
      X2 = 0.2D0
      F1 = FCN4(X1,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
    5 I = I + 1
C
C-------- LIMIT SEARCH TO INCREMENTS OF 100 INTERVALS --------------
C
      IF(I.GT.J*100) GOTO 10
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,X1,F1,X2,F2
      F2 = FCN4(X2,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
C
C-------- CHECK FOR FUNCTION VALUES OPPOSITE IN SIGN ---------------
C
      IF(F1*F2.GE.0.0D0) THEN
    X1 = X2
    F1 = F2
    IF(X2.LT.1.0D0) X2 = X2 + 0.1D0
        IF(X2.GE.1.0D0) X2 = X2 + 1.0D0
        IF(PASS.GE.1) PASS = PASS + 1
        GOTO 5
```

```
      ENDIF
C
C-------- IF INTERVAL FOUND, GO FIND ROOT --------------------------
C
      PRINT 201
      GOTO 20
C
C-------- IF NO INTERVAL FOUND, CONTINUE SEARCH ? ------------------
C
   10 PRINT 400,J*100
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- IF UNABLE TO FIND INTERVAL, LOOK AT FUNCTION VALUES ? -------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 15
      ENDIF
      IF(ANS.EQ.'N') THEN
    FLAG = 1
    RETURN
      ENDIF
      PASS = 1
      I = 0
      J = 1
      X1 = 0.1D0
      X2 = 0.2D0
      F1 = FCN4(X1,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      GOTO 5

C
C-------- GET ROOT IN INTERVAL: FIN HEIGHT AND TEMP AT TIP ----------
C
   20 CALL MDLIN4(FCN4,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      HT = XR
C
C-------- CALCULATE IDEAL HEAT DISSIPATED --------------------------
C
      QI = 2.0D0*SB*EMIS*HT*L*TB**4.0D0
C
```

```
C-------- CALCULATE FIN EFFICIENCY ---------------------------------
C
      EFF = Q/QI
C
C-------- SCALE OUTPUTS ---------------------------------------------
C
      HT = HT*0.01D0
      DEL0 = DEL0*0.01D0
      DELE = DELE*0.01D0
      L = L*0.01D0
      K = K*100.0D0
C
C-------- FORMAT ----------------------------------------------------
C
100   FORMAT('+','LOOKING IN INTERVAL: ',I3,'                        ')
200   FORMAT(/,' COMMENCING SEARCH FOR INTERVAL TO BRACKET ROOT.... ',/)
201   FORMAT(/,' INTERVAL FOUND, WILL NOW LOOK FOR ROOT ....          ',/)
300   FORMAT(' XL =',2X,D10.4,2X,'F(XL) =',2X,D10.4,2X,
     &       'XR =',2X,D10.4,2X,'F(XR) =',2X,D10.4)
400   FORMAT(/,' NO ROOT FOUND IN ',I3,' INTERVALS.  CONTINUE SEARCH'  ,
     & ' (Y OR N) ?'                                                   )
500   FORMAT(A2)
600   FORMAT(/,' **************** INCORRECT ENTRY ! ***************** ')
700   FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N)?     ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING. IF ',/,
     & ' F(XR) IS OF THE SAME SIGN AS F(XL) AND ITS ABSOLUTE VALUE  ',/,
     & ' IS INCREASING, THERE IS LITTLE PROBABILITY OF FINDING AN   ',/,
     & ' INTERVAL.  THE SAME GOES FOR VALUES OF F(XR) THAT OSCILLATE',/,
     & ' AROUND A CERTAIN VALUE.  FOR F(XR) TO BECOME OPPOSITE IN   ',/,
     & ' SIGN TO F(XR), IT MUST PASS THROUGH ZERO.                   ')
      RETURN
      END
C
C5C****** TPSY: FCN4(X,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L) ***************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE (BASE DERIV)
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = ABS*E
C   Q - HEAT INPUT THRU BASE
C   L - FIN LENGHT
C
C-------------------------------------------------------------------
```

```
C
      REAL*8 FUNCTION FCN4(X,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      REAL*8 X,X0,XFINAL,T0(2),TEND(2),F(2),H,TB,TE,HT,DEL0,DELE,
     &       K,K1,K2,Q,L,TSAVE,TOL
      EXTERNAL DERIV4
C
C-------- GUESS HEIGHT ----------------------------------------------
C
      HT = X
C
C-------- INITIALIZE LEFT END POINT BC'S: FCN AND DERIVATIVE KNOWN
C
      T0(1)= TB
      T0(2)= -Q/(L*DEL0*K)
C
C-------- INITIALIZE RIGHT END POINT BC'S TO 0
C
      TEND(1)=0.0D0
      TEND(2)=0.0D0
C
C-------- DEFINE PARAMETERS -----------------------------------------
C
      IF (X.LT.1.0D0) H = 0.05D0
      IF (X.GE.1.0D0) H = 0.1D0
      TOL = 0.0001D0
      X0 = 0.0D0
      XFINAL = HT
      TSAVE = 1000.0D0
C
C-------- USE RUNGE KUTTE FELHBERG TO SHOOT FROM LEFT END BC'S TO
C         RIGHT BC'S AS A FUNCTION OF FIN HEIGHT
C
10    IF(X0.LT.XFINAL+0.0001D0) THEN
    CALL RKFSY4(DERIV4,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
C
C-------- T INCREASING MEANS DIVERGENCE -----------------------------
C
    IF (TEND(1).GT.TSAVE) GOTO 20
    T0(1) = TEND(1)
    TSAVE = TEND(1)
    T0(2) = TEND(2)
    GO TO 10
      ENDIF
20    CONTINUE
      FCN4 = TEND(2)
      TE = TEND(1)
      RETURN
      END
C
C5D ***** TPSY: RKFSY4(DERIV4,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
```

```
C
C  PARAMETERS:
C
C  RKFSY4 - SUBROUTINE THAT SOLVES A SYSTEM OF 2 FIRST ORDER
C           DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-FEHLBERG
C           METHOD. THE EQUATIONS ARE OF THE FORM:
C
C           DT/DX = Y = F1(X,T)
C           DY/DX = F2(X,T,Y)
C
C  DERIV4 - A SUBROUTINE THAT COMPUTES VALUES OF THE 2 DERIVATIVES.
C
C  X0 - THE INITIAL VALUE OF INDEPENDENT VARIABLE
C  H - THE INCREMENT TO T, THE STEP SIZE
C  T0 - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  TEND - AN ARRAY THAT RETURNS THE FINAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  H - STEP SIZE
C  HT - HEIGHT OF FIN
C  DEL0 - THICKNESS AT BASE
C  DELE - THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  TOL - TOLERANCE
C  TWRK - AN ARRAY USED TO HOLD INTERMEDIATE VALUES DURING THE
C           COMPUTATION. IT MUST BE DIMENSIONED OF SIZE 6 X 2.
C
C-------------------------------------------------------------------
C
      SUBROUTINE RKFSY4(DERIV4,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
      REAL*8 X0,T0(2),TEND(2),F(2),TWRK(6,2),H,HT,DEL0,DELE,K,K1,K2,TOL,
     &     ERROR,SUM,STOREH,XEND
      INTEGER I
C
C-------- INITIALIZE FOR INTERVAL [X0, X0+H] ------------------------
C
      XEND = X0+H
      STOREH = H
C
C-------- CHECK TO SEE IF FINISHED ----------------------------------
C
1     IF(X0.GE.XEND) THEN
    H = STOREH
    RETURN
      ELSE
C
C-------- GET FIRST ESTIMATE ----------------------------------------
C
      CALL DERIV4(X0,T0,F,HT,DEL0,DELE,K,K1,K2)
```

```
      DO 10 I = 1,2
         TWRK(1,I) = H*F(I)
         TEND(I) = T0(I)+TWRK(1,I)/4.0D0
   10 CONTINUE
C
C-------- GET SECOND ESTIMATE --------------------------------------
C
      CALL DERIV4(X0+H/4.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 20 I = 1,2
         TWRK(2,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*3.0D0+TWRK(2,I)*9.0D0)/32.0D0
   20 CONTINUE
C
C-------- GET THIRD ESTIMATE ----------------------------------------
C
      CALL DERIV4(X0+3.0D0*H/8.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 30 I = 1,2
         TWRK(3,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*1932.0D0-TWRK(2,I)*7200.0D0
     &            + TWRK(3,I)*7296.0D0)/2197.0D0
   30 CONTINUE
C
C-------- GET FOURTH ESTIMATE --------------------------------------
C
      CALL DERIV4(X0+12.0*H/13.0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 40 I = 1,2
         TWRK(4,I) = H*F(I)
         TEND(I) = T0(I)+(439.0D0*TWRK(1,I)/216.0D0-8.0D0*TWRK(2,I)
     &    + 3680.0D0*TWRK(3,I)/513.0D0-845.0D0*TWRK(4,I)/4104.0D0)
   40 CONTINUE
C
C-------- GET FIFTH ESTIMATE ----------------------------------------
C
      CALL DERIV4(X0+H,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 50 I = 1,2
         TWRK(5,I) = H*F(I)
         TEND(I) = T0(I)-8.0D0*TWRK(1,I)/27.0+2.0D0*TWRK(2,I)
     &    - 3544.0D0*TWRK(3,I)/2565.0D0+1859.0D0*TWRK(4,I)/4104.0D0
     &    - 11.0D0*TWRK(5,I)/40.0D0
50    CONTINUE
C
C-------- GET SIXTH ESTIMATE ----------------------------------------
C
      CALL DERIV4(X0+H/2.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 60 I = 1,2
         TWRK(6,I) = H*F(I)
60    CONTINUE
C
C-------- ESTIMATE ERROR BY COMPUTING DIFFERENCE BETWEEN FOURTH AND
C         FIFTH ORDER EQUATIONS
```

```
C
      SUM = 0.0D0
      ERROR = 0.0D0
      DO 70 I = 1,2
         SUM = DABS(TWRK(1,I)/360.0D0-128.0D0*TWRK(3,I)/4275.0D0
     &         - 2197.0D0*TWRK(4,I)/75240.0D0+TWRK(5,I)/50.0D0
     &        + 2.0D0*TWRK(6,I)/55.0D0)
         ERROR = DMAX1(ERROR,SUM)
70    CONTINUE
C
C-------- IF ERROR LESS THAN TOLERANCE, COMPUTE X AT END OF ---------
C         INTERVAL FROM A WEIGHTED AVERAGE OF SIX ESTIMATES
C
      IF(ERROR.LT.TOL) THEN
    DO 80 I = 1,2
       TEND(I) = T0(I)+16.0D0*TWRK(1,I)/135.0D0
     &       + 6656.0D0*TWRK(3,I)/12825.0D0
     &       + 28561.0D0*TWRK(4,I)/56430.0-9.0D0*TWRK(5,I)/50.0D0
     &       + 2.0D0*TWRK(6,I)/55.0D0
       T0(I) = TEND(I)
80  CONTINUE
    X0 = X0+H
      ENDIF
C
C-------- IF ERROR GREATER THAN TOLERANCE, REDUCE STEP AND GO -------
C         AGAIN
C
      IF(ERROR.GT.TOL) THEN
    H = H/2.0D0
      ENDIF
C
C-------- IF ERROR IS SIGNIFICANLTY LESS THAN TOLERANCE, RELAX STEP
C
      IF(ERROR.LT.H*TOL/10.0D0) THEN
    H = H*2.0D0
      ENDIF
C
C-------- IF OVERSHOOT, REDUCE STEP ---------------------------------
C
      IF(X0+H.GT.XEND) THEN
    H = XEND-X0
      ENDIF
C
      ENDIF
C
C
      GO TO 1
      END
C
C5E****** TPSY: DERIV4(X,T,F,HT,DEL0,DELE,K,K1,K2) ******************
```

```
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE
C  T - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  HT - HEIGHT OF FIN
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-------------------------------------------------------------------
C
      SUBROUTINE DERIV4(X,T,F,HT,DEL0,DELE,K,K1,K2)
      REAL*8 X,T(2),F(2),HT,DEL0,DELE,K,K1,K2,DEL,DELP
      DEL(X) = DEL0+(DELE-DEL0)*X/HT
      DELP = (DELE-DEL0)/HT
      F(1) = T(2)
      F(2) = -(DELP*T(2))/DEL(X)+(K1*T(1)**4.0D0-K2)/(K*DEL(X))
      RETURN
      END
C
C5F ***** TPSY: MDLIN4(FCN4,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,HT,DEL0,
C                      DELE,K,K1,K2,Q,L)
C
C  PARAMETERS:
C
C  FCN4 - FUNCTION THAT COMPUTES VALUES FOR F. MUST BE DECLARED
C         EXTERNAL IN CALLING PROGRAM
C  X1,X2 - INITIAL VALUES OF X. F(X) MUST CHANGE SIGNS AT THESE
C          POINTS
C  XR - RETURNS THE ROOT TO THE MAIN PROGRAM
C  XTOL - TOLERANCE FOR X
C  FTOL - TOLERANCE FOR F
C  NLIM - LIMIT TO NUMBER OF ITERATIONS
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - BASE THICKNESS
C  DELE - TIP THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C   Q - HEAT INPUT THRU BASE
C   L - FIN LENGTH
C
C-------------------------------------------------------------------
C
```

```
      SUBROUTINE MDLIN4(FCN4,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      REAL*8 XERR,FSAVE,F1,F2,FR,X1,X2,XR,XTOL,FTOL,TB,TE,HT,DEL0,
     &       DELE,K,K1,K2,Q,L,FCN4,X1SAV,X2SAV,F1SAV,F2SAV
      INTEGER I,J,PASS
      CHARACTER*2 ANS
      DATA XTOL,FTOL/0.0001D0,0.00001D0/
C
C-------- INITIALIZE VALUES -------------------------------------------
C
      PRINT 200
      X1SAV = X1
      X2SAV = X2
      F1 = FCN4(X1,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      F2 = FCN4(X2,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      F1SAV = F1
      F2SAV = F2
      FSAVE = F1
C
C-------- INITIALIZE COUNTER ------------------------------------------
C
      I = 0
      J = 1
      PASS = 0
C
C-------- LIMIT SEARCH TO INCREMENTS OF 50 ITERATIONS -----------------
C
    5 I = I + 1
      IF(I.GT.J*50) GOTO 10
      XR = X2-F2*(X2-X1)/(F2-F1)
      FR = FCN4(XR,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,I,XR,FR
      IF(PASS.GE.1) PASS = PASS + 1
C
C-------- CHECK STOPPING CRITERIA -------------------------------------
C
      XERR = DABS(X1-X2)/2.0D0
      IF(XERR.LE.XTOL.OR.DABS(FR).LE.FTOL) RETURN
C
C-------- FIND NEW POINT ---------------------------------------------
C
      IF(FR*F1.GE.0.0D0) THEN
    X1 = XR
    F1 = FR
    IF(FR*FSAVE.GT.0.0D0) F2 = F2/2.0D0
    FSAVE = FR
      ELSE
    X2 = XR
    F2 = FR
```

```
     IF(FR*FSAVE.GT.0.0D0) F1 = F1/2.0D0
     FSAVE = FR
       ENDIF
       GOTO 5
C
C-------- FAIL TO FIND ROOT, CONTINUE SEARCH ? ---------------------
C
   10  PRINT 400,J*50
       READ 500,ANS
       IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
         PRINT 600
         GOTO 10
       ENDIF
       IF(ANS.EQ.'N') GOTO 15
       J = J + 1
       I = I - 1
       GOTO 5
C
C-------- FAIL TO FIND ROOT, LOOK AT FUNTION VALUES ? ----------------
C
   15  PRINT 700
       READ 500,ANS
       IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
         PRINT 600
     GOTO 15
       ENDIF
       IF(ANS.EQ.'N') STOP
       X1 = X1SAV
       X2 = X2SAV
       F1 = F1SAV
       F2 = F2SAV
       I = 0
       J = 1
       PASS = 1
       GOTO 5
C
C  ------- FORMAT -------------------------------------------- ---------
C
  100  FORMAT('+', 'ITERATION : ',I3,'                                ')
  200  FORMAT(/,' COMMENCING SEARCH FOR ROOT IN INTERVAL....           ',/)
  300  FORMAT(' AT ITERATION',I3,3X,' X = ',D15.6,3X,' F(X) = ',D15.6)
  400  FORMAT(/,' FAILED TO FIND ROOT AFTER ',I3,' ITERATIONS.'          ,
     & ' CONTINUE SEARCH (Y OR N) ?'                                     )
  500  FORMAT(A2)
  600  FORMAT(/' ***************** INCORRECT ANSWER ******************')
  700  FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N )?   ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING.    ',/,
     & ' IF F(XR) IS OSCILLATING ABOUT A CERTAIN VALUE OR ITS       ',/,
     & ' ABSOLUTE VALUE IS INCREASING THEN THERE IS LITTLE          ',/,
     & ' PROBABILITY THAT A ROOT WILL BE FOUND.                       ')
```

```
      END
C
C6A ***** TRAN HEADER5 ********************************************
C
C  SUBROUTINE TRAN(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
C
C-----------------------------------------------------------------------
C
C  FIN SHAPE:  TRIANGULAR
C
C  PROBLEM TYPE:  ANALYSIS
C
C  INPUT:  TB,L,HT,DEL0,K,ABS,EMIS,E
C
C  OUTPUT: TE,QI,Q,EFF
C
C-----------------------------------------------------------------------
C
C  PARAMETERS:
C
C  TB - TEMPERATURE AT THE BASE OF THE FIN
C  TE - TEMPERATURE AT THE TIP OF THE FIN
C  HT - HEIGHT OF THE FIN
C  L - LENGTH OF THE FIN
C  DEL0 - THICKNESS AT FIN BASE
C  DELE - THICKNESS AT FIN TIP: DELE = 0.01*DEL0
C  K - CONDUCTIVITY OF THE FIN
C  ABS - ABSORPTIVITY OF THE FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  QI - IDEAL HEAT DISSIPATED BY THE FIN
C  Q - REAL HEAT DISSIPATED BY THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  FLAG - 0 = CONVERGENCE, 1 = DIVERGENCE
C
C-----------------------------------------------------------------------
C
C  FUNCTIONS: FCN5(X,TB,TE,L,HT,DEL0,DELE,K,K1,K2,Q)
C
C   FUNCTION WHOSE ROOT IS FOUND (BASE DERIV & TIP TEMP)
C
C-----------------------------------------------------------------------
C
C  SUBROUTINES:
C
C  MDLIN4(FCN5,X1,X2,XR,XTOL,FTOL,NLIM,TB,HT,DEL,K,K1,K2):
C
C   FINDS THE ROOT OF AN EQUATION FCN4 BY THE METHOD OF MODIFIED
C   LINEAR INTERPOLATION
C
```

```
C  RKFSY5(DERIV5,XBEGIN,H,T0,TEND,F,HT,DEL0,DELE,K,K1,K2,TOL):
C
C    SOLVES SECOND ORDER NONLINEAR DE BY RUNGE-KUTTE-FEHLBERG METHOD
C
C  DERIV5(X,T,F,HT,DEL0,DELE,K,K1,K2):
C
C   COMPUTES DERIVATIVES FOR RKFSY5
C
C6B ***** TRAN MAIN5 *************************************************
C
      SUBROUTINE TRAN(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
      REAL*8 TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,X1,X2,XR,
     &       SB,TBDER,K1,K2,F1,F2,FCN5
      INTEGER I,J,PASS,FLAG
      CHARACTER*2 ANS
      EXTERNAL FCN5
C
C-------- DEFINE CONSTANTS -----------------------------------------
C
      SB = 5.67051D-12
      K1 = 2.0D0*SB*EMIS
      K2 = E*ABS
C
C-------- SCALE INPUTS ---------------------------------------------
C
      HT = HT*100.0D0
      DEL0 = DEL0*100.0D0
      L = L*100.0D0
      K = K*0.01D0
C
C-------- FOR TRIANGULAR FIN: TIP THICKNESS = 0.01D0 X BASE THICKNESS
C
      DELE = DEL0*0.01D0
C
C-------- START COUNTERS -------------------------------------------
C
      PASS = 0
      I = 0
      J = 1
C
C-------- START SEARCH WITH INTERVAL [-1,0] ------------------------
C
      PRINT 200
      X1 = 0.0D0
      X2 = -1.0D0
      F1 = FCN5(X1,TB,TE,HT,DEL0,DELE,K,K1,K2)
    5 I = I + 1
C
C-------- LIMIT SEARCH TO INCREMENTS OF 100 INTERVALS --------------
C
```

```
      IF(I.GT.J*100) GOTO 10
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,X1,F1,X2,F2
      F2 = FCN5(X2,TB,TE,HT,DEL0,DELE,K,K1,K2)
C
C-------- CHECK FOR FUNCTION VALUES OPPOSITE IN SIGN ----------------
C
      IF(F1*F2.GE.0.0D0) THEN
        X1 = X2
        F1 = F2
        X2 = X2 - 1.0D0
        IF(PASS.GE.1) PASS = PASS + 1
        GOTO 5
      ENDIF
C
C-------- IF INTERVAL FOUND, GO FIND ROOT ---------------------------
C
      PRINT 201
      GOTO 20
C
C-------- IF NO INTERVAL FOUND, CONTINUE SEARCH ? -------------------
C
   10 PRINT 400,J*100
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- IF UNABLE TO FIND INTERVAL, LOOK AT FUNCTION VALUES ? -------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 15
      ENDIF
      IF(ANS.EQ.'N') THEN
    FLAG = 1
    RETURN
      ENDIF
      PASS = 1
      I = 0
      J = 1
      X1 = 0.0D0
```

```
      X2 = -1.0D0
      F1 = FCN5(X1,TB,TE,HT,DEL0,DELE,K,K1,K2)
      GOTO 5
C
C-------- GET ROOT IN INTERVAL: DERIV AT BASE AND TEMP AT TIP -------
C
   20 CALL MDLIN5(FCN5,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2)
      TBDER = XR
C
C-------- CALCULATE IDEAL HEAT DISSIPATED ---------------------------
C
      QI = 2.0D0*SB*EMIS*HT*L*TB**4.0D0
C
C--------- CALCULATE REAL*8 HEAT DISSIPATED ---------------------------
C
      Q = -K*DEL0*L*TBDER
C
C-------- CALCULATE FIN EFFICIENCY ----------------------------------
C
      EFF = Q/QI
C
C
C-------- SCALE OUTPUTS ---------------------------------------------
C
      DEL0 = DEL0*0.01D0
      DELE = DELE*0.01D0
      L = L*0.01D0
      HT = HT*0.01D0
      K = K*100.0D0
      RETURN
C
C-------- FORMAT ----------------------------------------------------
C
100   FORMAT('+','LOOKING IN INTERVAL: ',I3,'                   ')
200   FORMAT(/,' COMMENCING SEARCH FOR INTERVAL TO BRACKET ROCT.... ',/)
201   FORMAT(/,' INTERVAL FOUND, WILL NOW LOOK FOR ROOT ....         ',/)
300   FORMAT(' XL =',2X,D10.4,2X,'F(XL) =',2X,D10.4,2X,
     &        'XR =',2X,D10.4,2X,'F(XR) =',2X,D10.4)
400   FORMAT(/,' NO ROOT FOUND IN ',I3,' INTERVALS.  CONTINUE SEARCH'  ,
     & ' (Y OR N) ?'                                                   )
500   FORMAT(A2)
600   FORMAT(/,' **************** INCORRECT ENTRY ! ***************** ')
700   FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N)?     ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING. IF ',/,
     & ' F(XR) IS OF THE SAME SIGN AS F(XL) AND ITS ABSOLUTE VALUE  ',/,
     & ' IS INCREASING, THERE IS LITTLE PROBABILITY OF FINDING AN   ',/,
     & ' INTERVAL.  THE SAME GOES FOR VALUES OF F(XR) THAT OSCILLATE',/,
     & ' AROUND A CERTAIN VALUE.  FOR F(XR) TO BECOME OPPOSITE IN   ',/,
     & ' SIGN TO F(XR), IT MUST PASS THROUGH ZERO.                   ')
```

```
      END
C
C6C****** TRAN: FCN5(X,TB,TE,HT,DEL0,DELE,K,K1,K2) ******************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE (BASE DERIV)
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = ABS*E
C
C-------------------------------------------------------------------
C
      REAL*8 FUNCTION FCN5(X,TB,TE,HT,DEL0,DELE,K,K1,K2)
      REAL*8 X,X0,XFINAL,T0(2),TEND(2),F(2),H,TB,TE,HT,DEL0,DELE,
     &       K,K1,K2,TSAVE,TOL
      EXTERNAL DERIV5
C
C-------- INITIALIZE LEFT END POINT BC'S: KNOW FCT VALUE, GUESS
C         DERIVATIVE
C
      T0(1)= TB
      T0(2)= X
C
C-------- INITIALIZE RIGHT END POINT BC'S TO 0 ---------------------
C
      TEND(1)=0.0D0
      TEND(2)=0.0D0
C
C-------- DEFINE PARAMETERS -----------------------------------------
C
      H= 0.1D0
      TOL = 0.0001D0
      X0 = 0.0D0
      XFINAL = HT
      TSAVE = 1000.0D0
C
C-------- USE RUNGE KUTTE FELHBERG TO SHOOT FROM LEFT END BC'S TO
C         RIGHT BC'S AS A FUNCTION OF BASE DERIV
C
10    IF(X0.LT.XFINAL+0.0001D0) THEN
    CALL RKFSY5(DERIV5,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
C
C-------- INCREASING T MEANS DIVERGENCE ----------------------------
C
```

```
    IF (TEND(1).GE.TSAVE) GOTO 20
    T0(1) = TEND(1)
    TSAVE = TEND(1)
    T0(2) = TEND(2)
    GO TO 10
      ENDIF
20    CONTINUE
      FCN5 = TEND(2)
      TE = TEND(1)
      RETURN
      END
C
C6D ***** TRAN: RKFSY5(DERIV5,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
C
C  PARAMETERS:
C
C  RKFSY5 - SUBROUTINE THAT SOLVES A SYSTEM OF 2 FIRST ORDER
C           DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-FEHLBERG
C           METHOD. THE EQUATIONS ARE OF THE FORM:
C
C           DT/DX = Y = F1(X,T)
C           DY/DX = F2(X,T,Y)
C
C  DERIV5 - A SUBROUTINE THAT COMPUTES VALUES OF THE 2 DERIVATIVES.
C
C  X0 - THE INITIAL VALUE OF INDEPENDENT VARIABLE
C  T0 - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  TEND - AN ARRAY THAT RETURNS THE FINAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  H - STEP SIZE
C  HT - HEIGHT OF FIN
C  DEL0 - THICKNESS AT BASE
C  DELE - THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  TOL - TOLERANCE
C  TWRK - AN ARRAY USED TO HOLD INTERMEDIATE VALUES DURING THE
C     COMPUTATION. IT MUST BE DIMENSIONED OF SIZE 6 X 2.
C
C--------------------------------------------------------------------
C
      SUBROUTINE RKFSY5(DERIV5,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
      REAL*8 X0,T0(2),TEND(2),F(2),TWRK(6,2),H,HT,DEL0,DELE,K,K1,K2,
     &       TOL,ERROR,SUM,STOREH,XEND
      INTEGER I
C
C-------- INITIALIZE FOR INTERVAL [X0, X0+H] -----------------------
C
      XEND = X0+H
```

```
      STOREH = H
C
C-------- CHECK TO SEE IF WE ARE FINISHED --------------------------
C
1     IF(X0.GE.XEND) THEN
    H = STOREH
    RETURN
      ELSE
C
C-------- GET FIRST ESTIMATE OF THE DELTA X'S -----------------------
C
      CALL DERIV5(X0,T0,F,HT,DEL0,DELE,K,K1,K2)
      DO 10 I = 1,2
         TWRK(1,I) = H*F(I)
         TEND(I) = T0(I)+TWRK(1,I)/4.0D0
   10 CONTINUE
C
C-------- GET SECOND ESTIMATE ---------------------------------------
C
      CALL DERIV5(X0+H/4.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 20 I = 1,2
         TWRK(2,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*3.0D0+TWRK(2,I)*9.0D0)/32.0D0
   20 CONTINUE
C
C-------- GET THIRD ESTIMATE ----------------------------------------
C
      CALL DERIV5(X0+3.0D0*H/8.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 30 I = 1,2
         TWRK(3,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*1932.0D0-TWRK(2,I)*7200.0D0
     &           + TWRK(3,I)*7296.0D0)/2197.0D0
   30 CONTINUE
C
C-------- GET FOURTH ESTIMATE ---------------------------------------
C
      CALL DERIV5(X0+12.0*H/13.0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 40 I = 1,2
         TWRK(4,I) = H*F(I)
         TEND(I) = T0(I)+(439.0D0*TWRK(1,I)/216.0D0-8.0D0*TWRK(2,I)
     &    + 3680.0D0*TWRK(3,I)/513.0D0-845.0D0*TWRK(4,I)/4104.0D0)
   40 CONTINUE
C
C-------- GET FIFTH ESTIMATE ----------------------------------------
C
      CALL DERIV5(X0+H,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 50 I = 1,2
         TWRK(5,I) = H*F(I)
         TEND(I) = T0(I)-8.0D0*TWRK(1,I)/27.0+2.0D0*TWRK(2,I)
     &    - 3544.0D0*TWRK(3,I)/2565.0D0+1859.0D0*TWRK(4,I)/4104.0D0
```

```
     &     - 11.0D0*TWRK(5,I)/40.0D0
50     CONTINUE
C
C-------- GET SIXTH ESTIMATE ----------------------------------------
C
       CALL DERIV5(X0+H/2.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
       DO 60 I = 1,2
          TWRK(6,I) = H*F(I)
60     CONTINUE
C
C-------- ESTIMATE THE ERROR BY COMPUTING THE DIFFERENCE BETWEEN
C          THE FOURTH AND FIFTH ORDER EQUATIONS
C
       SUM = 0.0D0
       ERROR = 0.0D0
       DO 70 I = 1,2
          SUM = DABS(TWRK(1,I)/360.0D0-128.0D0*TWRK(3,I)/4275.0D0
     &          - 2197.0D0*TWRK(4,I)/75240.0D0+TWRK(5,I)/50.0D0
     &          + 2.0D0*TWRK(6,I)/55.0D0)
          ERROR = DMAX1(ERROR,SUM)
70     CONTINUE
C
C-------- IF ERROR LESS THAN TOLERANCE, COMPUTE X AT END OF ---------
C          THE INTERVAL FROM A WEIGHTED AVERAGE OF SIX ESTIMATES,
C          THEN RETURN.
C
       IF(ERROR.LT.TOL) THEN
     DO 80 I = 1,2
        TEND(I) = T0(I)+16.0D0*TWRK(1,I)/135.0D0
     &        + 6656.0D0*TWRK(3,I)/12825.0D0
     &        + 28561.0D0*TWRK(4,I)/56430.0-9.0D0*TWRK(5,I)/50.0D0
     &        + 2.0D0*TWRK(6,I)/55.0D0
        T0(I) = TEND(I)
80   CONTINUE
     X0 = X0+H
       ENDIF
C
C-------- IF ERROR GREATER THAN TOLERANCE, THEN REDUCE STEP ---------
C
       IF(ERROR.GT.TOL) THEN
     H = H/2.0D0
       ENDIF
C
C-------- IF ERROR IS SIGNIFICANTLY LESS THAN TOLERANCE, RELAX STEP
C
       IF(ERROR.LT.H*TOL/10.0D0) THEN
     H = H*2.0D0
       ENDIF
C
C-------- IF OVERSHOOT, REDUCE STEP ---------------------------------
```

```
C
      IF(X0+H.GT.XEND) THEN
    H = XEND-X0
      ENDIF
C
      ENDIF
C
      GO TO 1
      END
C
C
C6E****** TRAN: DERIV5(X,T,F,HT,DEL0,DELE,K,K1,K2) ******************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE
C  T - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  HT - HEIGHT OF FIN
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0D0*SB*EMIS
C  K2 - CONSTANT = E*DABS
C
C---------------------------------------------------------------------
C
      SUBROUTINE DERIV5(X,T,F,HT,DEL0,DELE,K,K1,K2)
      REAL*8 X,T(2),F(2),HT,DEL0,DELE,K,K1,K2,DEL,DELP
      DEL(X) = DEL0+(DELE-DEL0)*X/HT
      DELP = (DELE-DEL0)/HT
      F(1) = T(2)
      F(2) = -(DELP*T(2))/DEL(X)+(K1*T(1)**4.0D0-K2)/(K*DEL(X))
      RETURN
      END
C
C6F ***** TRAN: MDLIN5(FCN5,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,HT,DEL0,
C                      DELE,K,K1,K2)
C
C  PARAMETERS:
C
C  FCN5 - FUNCTION THAT COMPUTES VALUES FOR F. MUST BE DECLARED
C         EXTERNAL IN CALLING PROGRAM
C  X1,X2 - INITIAL VALUES OF X. F(X) MUST CHANGE SIGNS AT THESE
C          POINTS
C  XR - RETURNS THE ROOT TO THE MAIN PROGRAM
C  XTOL - TOLERANCE FOR X
C  FTOL - TOLERANCE FOR F
C  NLIM - LIMIT TO NUMBER OF ITERATIONS
C  TB - BASE TEMPERATURE
```

```
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - BASE THICKNESS
C  DELE - TIP THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-----------------------------------------------------------------------
C
      SUBROUTINE MDLIN5(FCN5,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2)
      REAL*8 XERR,FSAVE,F1,F2,FR,X1,X2,XR,XTOL,FTOL,TB,TE,HT,DEL0,
     &       DELE,K,K1,K2,FCN5,X1SAV,X2SAV,F1SAV,F2SAV
      INTEGER I,J,PASS
      CHARACTER*2 ANS
      DATA XTOL,FTOL/0.0001D0,0.00001D0/
C
C-------- INITIALIZE VALUES ------------------------------------------
C
      PRINT 200
      X1SAV = X1
      X2SAV = X2
      F1 = FCN5(X1,TB,TE,HT,DEL0,DELE,K,K1,K2)
      F2 = FCN5(X2,TB,TE,HT,DEL0,DELE,K,K1,K2)
      F1SAV = F1
      F2SAV = F2
      FSAVE = F1
C
C-------- INITIALIZE COUNTER -----------------------------------------
C
      I = 0
      J = 1
      PASS = 0
C
C-------- LIMIT SEARCH TO INCREMENTS OF 50 ITERATIONS ----------------
C
    5 I = I + 1
      IF(I.GT.J*50) GOTO 10
      XR = X2-F2*(X2-X1)/(F2-F1)
      FR = FCN5(XR,TB,TE,HT,DEL0,DELE,K,K1,K2)
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,I,XR,FR
      IF(PASS.GE.1) PASS = PASS + 1
C
C-------- CHECK STOPPING CRITERIA ----------------------------------
C
      XERR = DABS(X1-X2)/2.0D0
      IF(XERR.LE.XTOL.OR.DABS(FR).LE.FTOL) RETURN
C
```

```
C-------- FIND NEW POINT ------------------------------------------
C
      IF(FR*F1.GE.0.0D0) THEN
    X1 = XR
    F1 = FR
    IF (FR*FSAVE.GT.0.0D0) F2 = F2/2.0D0
    FSAVE = FR
      ELSE
    X2 = XR
    F2 = FR
    IF (FR*FSAVE.GT.0.0D0) F1 = F1/2.0D0
    FSAVE = FR
      ENDIF
      GOTO 5
C
C-------- FAIL TO FIND ROOT, CONTINUE SEARCH ? ----------------------
C
   10 PRINT 400,J*50
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- FAIL TO FIND ROOT, LOOK AT FUNTION VALUES ? ----------------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
    GOTO 15
      ENDIF
      IF(ANS.EQ.'N') STOP
      X1 = X1SAV
      X2 = X2SAV
      F1 = F1SAV
      F2 = F2SAV
      I = 0
      J = 1
      PASS = 1
      GOTO 5
C
C  ----------------------------------------------------------------------
C
  100 FORMAT('+', 'ITERATION : ',I3,'                              ')
  200 FORMAT(/,' COMMENCING SEARCH FOR ROOT IN INTERVAL....          ',/)
```

```
  300 FORMAT(' AT ITERATION',I3,3X,' X = ',D15.6,3X,' F(X) = ',D15.6)
  400 FORMAT(/,' FAILED TO FIND ROOT AFTER ',I3,' ITERATIONS.'          ,
     & ' CONTINUE SEARCH (Y OR N) ?'                                    )
  500 FORMAT(A2)
  600 FORMAT(/' ***************** INCORRECT ANSWER ******************')
  700 FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N )?   ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING.    ',/,
     & ' IF F(XR) IS OSCILLATING ABOUT A CERTAIN VALUE OR ITS       ',/,
     & ' ABSOLUTE VALUE IS INCREASING THEN THERE IS LITTLE          ',/,
     & ' PROBABILITY THAT A ROOT WILL BE FOUND.                      ')

      END
C
C7A ***** TRSY HEADER6 *************************************************
C
C  SUBROUTINE TRSY(TB,TE,HT,L,DEL0,DELE,K,DABS,EMIS,E,QI,Q,EFF,FLAG)
C
C----------------------------------------------------------------------
C
C  FIN SHAPE:  TRIANGULAR
C
C  PROBLEM TYPE:  SYNTHESIS
C
C  INPUT:  TB,L,DEL0,K,ABS,EMIS,E,Q
C
C  OUTPUT: TE,HT,QI,EFF
C
C----------------------------------------------------------------------
C
C  PARAMETERS:
C
C  TB - TEMPERATURE AT THE BASE OF THE FIN
C  TE - TEMPERATURE AT THE TIP OF THE FIN
C  HT - HEIGHT OF THE FIN
C  L - LENGTH OF THE FIN
C  DEL0 - THICKNESS AT FIN BASE
C  DELE - THICKNESS AT FIN TIP: DELE = 0.01D0*DEL0
C  K - CONDUCTIVITY OF THE FIN
C  DABS - ABSORPTIVITY OF THE FIN
C  EMIS - EMISSIVITY OF THE FIN
C  E - EXTERNAL HEAT INCIDENT ON THE FIN
C  QI - IDEAL HEAT DISSIPATED BY THE FIN
C  Q - REAL HEAT DISSIPATED BY THE FIN
C  EFF - EFFICIENCY OF THE FIN
C  FLAG - 0 = CONVERGENCE, 1 = DIVERGENCE
C
C----------------------------------------------------------------------
C
C  FUNCTIONS: FCN6(X,TB,TE,L,HT,DEL0,DELE,K,K1,K2,Q)
C
```

```
C   FUNCTION WHOSE ROOT IS FOUND (BASE DERIV & TIP TEMP)
C
C----------------------------------------------------------------------
C
C  SUBROUTINES:
C
C  MDLIN6(FCN6,X1,X2,XR,XTOL,FTOL,NLIM,TB,HT,DEL,K,K1,K2):
C
C   FINDS THE ROOT OF AN EQUATION FCN4 BY THE METHOD OF MODIFIED
C   LINEAR INTERPOLATION
C
C  RKFSY6(DERIV6,XBEGIN,H,T0,TEND,F,HT,DEL0,DELE,K,K1,K2,TOL):
C
C    SOLVES SECOND ORDER NONLINEAR DE BY RUNGE-KUTTE-FEHLBERG METHOD
C
C  DERIV6(X,T,F,HT,DEL0,DELE,K,K1,K2):
C
C   COMPUTES DERIVATIVES FOR RKFSY6
C

C7B ***** TRSY MAIN6 **************************************************
C
      SUBROUTINE TRSY(TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,FLAG)
      REAL*8 TB,TE,L,HT,DEL0,DELE,K,ABS,EMIS,E,QI,Q,EFF,X1,X2,XR,
     &       SB,K1,K2,F1,F2,FCN6
      INTEGER I,J,PASS,FLAG
      CHARACTER*2 ANS
      EXTERNAL FCN6
C
C-------- DEFINE CONSTANTS -------------------------------------------
C
      SB = 5.67051D-12
      K1 = 2.0D0*SB*EMIS
      K2 = E*ABS
C
C-------- SCALE INPUTS -----------------------------------------------
C
      DEL0 = DEL0*100.0D0
      L = L*100.0D0
      K = K*0.01D0
C
C-------- FOR TRIANGULAR FIN: TIP THICKNESS = 0.01D0 BASE THICKNESS --
C
      DELE = DEL0*0.01D0
C
C-------- START COUNTERS ---------------------------------------------
C
      PASS = 0
      I = 0
      J = 1
```

```
C
C-------- START SEARCH WITH INTERVAL [0.1,0.2] ---------------------
C
      PRINT 200
      X1 = 1.0D0
      X2 = 2.0D0
      F1 = FCN6(X1,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
    5 I = I + 1
C
C-------- LIMIT SEARCH TO INCREMENTS OF 100 INTERVALS ---------------
C
      IF(I.GT.J*100) GOTO 10
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,X1,F1,X2,F2
      F2 = FCN6(X2,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
C
C-------- CHECK FOR FUNCTION VALUES OPPOSITE IN SIGN ----------------
C
      IF(F1*F2.GE.0.0D0) THEN
        X1 = X2
        F1 = F2
    X2 = X2 + 1.0D0
        IF(PASS.GE.1) PASS = PASS + 1
        GOTO 5
      ENDIF
C
C-------- IF INTERVAL FOUND, GO FIND ROOT ---------------------------
C
      PRINT 201
      GOTO 20
C
C-------- IF NO INTERVAL FOUND, CONTINUE SEARCH ? -------------------
C
   10 PRINT 400,J*100
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- IF UNABLE TO FIND INTERVAL, LOOK AT FUNCTION VALUES ? -------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
```

```
          PRINT 600
          GOTO 15
        ENDIF
        IF(ANS.EQ.'N') THEN
      FLAG = 1
      RETURN
        ENDIF
        PASS = 1
        I = 0
        J = 1
        X1 = 1.0D0
        X2 = 2.0D0
        F1 = FCN6(X1,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
        GOTO 5
C
C-------- GET ROOT IN INTERVAL: FIN HEIGHT AND TEMP AT TIP ----------
C
   20 CALL MDLIN6(FCN6,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
        HT = XR
C
C-------- CALCULATE IDEAL HEAT DISSIPATED --------------------------
C
        QI = 2.0D0*SB*EMIS*HT*L*TB**4.0D0
C
C-------- CALCULATE FIN EFFICIENCY ---------------------------------
C
        EFF = Q/QI
C
C-------- SCALE OUTPUTS ------------------------------------------
C
        HT = HT*0.01D0
        DEL0 = DEL0*0.01D0
        L = L*0.01D0
        K = K*100.0D0
        RETURN
C
C-------- FORMAT -------------------------------------------------
C
100   FORMAT('+','LOOKING IN INTERVAL: ',I3,'                         ')
200   FORMAT(/,' COMMENCING SEARCH FOR INTERVAL TO BRACKET ROOT.... ',/)
201   FORMAT(/,' INTERVAL FOUND, WILL NOW LOOK FOR ROOT ....        ',/)
300   FORMAT(' XL =',2X,D10.4,2X,'F(XL) =',2X,D10.4,2X,
     &        'XR =',2X,D10.4,2X,'F(XR) =',2X,D10.4)
400   FORMAT(/,' NO ROOT FOUND IN ',I3,' INTERVALS.  CONTINUE SEARCH'  ,
     & ' (Y OR N) ?'                                                    )
500   FORMAT(A2)
600   FORMAT(/,' **************** INCORRECT ENTRY ! **************** ')
700   FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N)?     ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING. IF ',/,
     & ' F(XR) IS OF THE SAME SIGN AS F(XL) AND ITS ABSOLUTE VALUE  ',/,
```

```
     & ' IS INCREASING, THERE IS LITTLE PROBABILITY OF FINDING AN   ',/,
     & ' INTERVAL.  THE SAME GOES FOR VALUES OF F(XR) THAT OSCILLATE',/,
     & ' AROUND A CERTAIN VALUE.  FOR F(XR) TO BECOME OPPOSITE IN   ',/,
     & ' SIGN TO F(XR), IT MUST PASS THROUGH ZERO.                  ')
      END
C
C7C****** TRSY: FCN6(X,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L) ***************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE (BASE DERIV)
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = ABS*E
C  Q - HEAT INPUT THRU BASE
C  L - FIN LENGTH
C
C-------------------------------------------------------------------
C
      REAL*8 FUNCTION FCN6(X,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      REAL*8 X,X0,XFINAL,T0(2),TEND(2),F(2),H,TB,TE,HT,DEL0,DELE,
     &       K,K1,K2,Q,L,TSAVE,TOL
      EXTERNAL DERIV6
C
C-------- GUESS HEIGHT ---------------------------------------------
C
      HT = X
C
C-------- INITIALIZE LEFT END POINT BC'S: FCN AND DERIVATIVE KNOWN
C
      T0(1)= TB
      T0(2)= -Q/(L*DEL0*K)
C
C-------- INITIALIZE RIGHT END POINT BC'S TO 0 ---------------------
C
      TEND(1)=0.0D0
      TEND(2)=0.0D0
C
C-------- DEFINE PARAMETERS -----------------------------------------
C
C     IF (X.LT.1.0D0) H = 0.01D0
C     IF (X.GE.1.0D0) H = 0.1D0
      H = 0.01D0
      TOL = 0.0001D0
      X0 = 0.0D0
```

```
      XFINAL = X
      TSAVE = 1000.0D0
C
C-------- USE RUNGE KUTTE FELHBERG TO SHOOT FROM LEFT END BC'S TO
C         RIGHT BC'S AS A FUNCTION OF FIN HEIGHT
C
10    IF(X0.LT.XFINAL+0.0001D0) THEN
    CALL RKFSY6(DERIV6,X0,T0,TEND,F,H,HT,DEL0,DELE,Γ,K1,K2,TOL)
C
C-------- T INCREASING MEANS DIVERGENCE ----------------------------
C
    IF (TEND(1).GT.TSAVE) GOTO 20
    T0(1) = TEND(1)
    TSAVE = TEND(1)
    T0(2) = TEND(2)
    GO TO 10
      ENDIF
20    CONTINUE
      FCN6 = TEND(2)
      TE = TEND(1)
      RETURN
      END
C
C7D ***** TRSY: RKFSY6(DERIV6,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
C
C  PARAMETERS:
C
C  RKFSY6 - SUBROUTINE THAT SOLVES A SYSTEM OF 2 FIRST ORDER
C           DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-FEHLBERG
C           METHOD. THE EQUATIONS ARE OF THE FORM:
C
C           DT/DX = Y = F1(X,T)
C           DY/DX = F2(X,T,Y)
C
C  DERIV6 - A SUBROUTINE THAT COMPUTES VALUES OF THE 2 DERIVATIVES.
C
C  X0 - THE INITIAL VALUE OF INDEPENDENT VARIABLE
C  T0 - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  TEND - AN ARRAY THAT RETURNS THE FINAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  H - STEP SIZE
C  HT - HEIGHT OF FIN
C  DEL0 - THICKNESS AT BASE
C  DELE - THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  TOL - TOLERANCE
C  TWRK - AN ARRAY USED TO HOLD INTERMEDIATE VALUES DURING THE
C           COMPUTATION. IT MUST BE DIMENSIONED OF SIZE 6 X 2.
```

```
C
C-----------------------------------------------------------------------
C
      SUBROUTINE RKFSY6(DERIV6,X0,T0,TEND,F,H,HT,DEL0,DELE,K,K1,K2,TOL)
      REAL*8 X0,T0(2),TEND(2),F(2),TWRK(6,2),H,HT,DEL0,DELE,K,K1,K2,TOL,
     &       ERROR,SUM,STOREH,XEND
      INTEGER I
C
C-------- INITIALIZE FOR INTERVAL [X0, X0+H] -----------------------
C
      XEND = X0+H
      STOREH = H
C
C-------- CHECK TO SEE IF FINISHED ----------------------------------
C
1     IF(X0.GE.XEND) THEN
    H = STOREH
    RETURN
      ELSE
C
C-------- GET FIRST ESTIMATE ---------------------------------------
C
      CALL DERIV6(X0,T0,F,HT,DEL0,DELE,K,K1,K2)
      DO 10 I = 1,2
         TWRK(1,I) = H*F(I)
         TEND(I) = T0(I)+TWRK(1,I)/4.0D0
   10 CONTINUE
C
C-------- GET SECOND ESTIMATE --------------------------------------
C
      CALL DERIV6(X0+H/4.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 20 I = 1,2
         TWRK(2,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*3.0D0+TWRK(2,I)*9.0D0)/32.0D0
   20 CONTINUE
C
C-------- GET THIRD ESTIMATE ---------------------------------------
C
      CALL DERIV6(X0+3.0D0*H/8.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 30 I = 1,2
         TWRK(3,I) = H*F(I)
         TEND(I) = T0(I)+(TWRK(1,I)*1932.0D0-TWRK(2,I)*7200.0D0
     &             + TWRK(3,I)*7296.0D0)/2197.0D0
   30 CONTINUE
C
C-------- GET FOURTH ESTIMATE --------------------------------------
C
      CALL DERIV6(X0+12.0*H/13.0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 40 I = 1,2
         TWRK(4,I) = H*F(I)
```

```
         TEND(I) = T0(I)+(439.0D0*TWRK(1,I)/216.0D0-8.0D0*TWRK(2,I)
     &    + 3680.0D0*TWRK(3,I)/513.0D0-845.0D0*TWRK(4,I)/4104.0D0)
   40 CONTINUE
C
C-------- GET FIFTH ESTIMATE ---------------------------------------
C
      CALL DERIV6(X0+H,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 50 I = 1,2
         TWRK(5,I) = H*F(I)
         TEND(I) = T0(I)-8.0D0*TWRK(1,I)/27.0+2.0D0*TWRK(2,I)
     &    - 3544.0D0*TWRK(3,I)/2565.0D0+1859.0D0*TWRK(4,I)/4104.0D0
     &    - 11.0D0*TWRK(5,I)/40.0D0
50    CONTINUE
C
C-------- GET SIXTH ESTIMATE ---------------------------------------
C
      CALL DERIV6(X0+H/2.0D0,TEND,F,HT,DEL0,DELE,K,K1,K2)
      DO 60 I = 1,2
         TWRK(6,I) = H*F(I)
60    CONTINUE
C
C-------- ESTIMATE ERROR BY COMPUTING DIFFERENCE BETWEEN FOURTH AND
C         FIFTH ORDER EQUATIONS
C
      SUM = 0.0D0
      ERROR = 0.0D0
      DO 70 I = 1,2
         SUM = DABS(TWRK(1,I)/360.0D0-128.0D0*TWRK(3,I)/4275.0D0
     &         - 2197.0D0*TWRK(4,I)/75240.0D0+TWRK(5,I)/50.0D0
     &         + 2.0D0*TWRK(6,I)/55.0D0)
         ERROR = DMAX1(ERROR,SUM)
70    CONTINUE
C
C-------- IF ERROR LESS THAN TOLERANCE, COMPUTE X AT END OF ---------
C         INTERVAL FROM A WEIGHTED AVERAGE OF SIX ESTIMATES
C
      IF (ERROR.LT.TOL) THEN
         DO 80 I = 1,2
            TEND(I) = T0(I)+16.0D0*TWRK(1,I)/135.0D0
     &         + 6656.0D0*TWRK(3,I)/12825.0D0
     &         + 28561.0D0*TWRK(4,I)/56430.0-9.0D0*TWRK(5,I)/50.0D0
     &         + 2.0D0*TWRK(6,I)/55.0D0
            T0(I) = TEND(I)
80       CONTINUE
         X0 = X0+H
      ENDIF
C
C-------- IF ERROR GREATER THAN TOLERANCE, REDUCE STEP AND GO -------
C         AGAIN
C
```

```
      IF(ERROR.GT.TOL) THEN
    H = H/2.0D0
      ENDIF
C
C-------- IF ERROR IS SIGNIFICANLTY LESS THAN TOLERANCE, RELAX STEP
C
      IF(ERROR.LT.H*TOL/10.0D0) THEN
    H = H*2.0D0
      ENDIF
C
C-------- IF OVERSHOOT, REDUCE STEP --------------------------------
C
      IF(X0+H.GT.XEND) THEN
    H = XEND-X0
      ENDIF
C
      ENDIF
C
C
      GO TO 1
      END
C
C7E****** TRSY: DERIV6(X,T,F,HT,DEL0,DELE,K,K1,K2) *****************
C
C  PARAMETERS:
C
C  X - INDEPENDENT VARIABLE
C  T - THE ARRAY THAT HOLDS THE INITIAL VALUES OF THE FUNCTIONS
C  F - AN ARRAY THAT HOLDS VALUES OF THE DERIVATIVES
C  HT - HEIGHT OF FIN
C  DEL0 - FIN THICKNESS AT BASE
C  DELE - FIN THICKNESS AT TIP
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C
C-----------------------------------------------------------------
C
      SUBROUTINE DERIV6(X,T,F,HT,DEL0,DELE,K,K1,K2)
      REAL*8 X,T(2),F(2),HT,DEL0,DELE,K,K1,K2,DEL,DELP
      DEL(X) = DEL0+(DELE-DEL0)*X/HT
      DELP = (DELE-DEL0)/HT
      F(1) = T(2)
      F(2) = -(DELP*T(2))/DEL(X)+(K1*T(1)**4.0D0-K2)/(K*DEL(X))
      RETURN
      END
C
C7F ***** TRSY: MDLIN6(FCN6,X1,X2,XR,XTOL,FTOL,NLIM,TB,TE,HT,DEL0,
C                       DELE,K,K1,K2,Q,L)
C
```

```
C  PARAMETERS:
C
C  FCN6 - FUNCTION THAT COMPUTES VALUES FOR F. MUST BE DECLARED
C         EXTERNAL IN CALLING PROGRAM
C  X1,X2 - INITIAL VALUES OF X. F(X) MUST CHANGE SIGNS AT THESE
C          POINTS
C  XR - RETURNS THE ROOT TO THE MAIN PROGRAM
C  XTOL - TOLERANCE FOR X
C  FTOL - TOLERANCE FOR F
C  NLIM - LIMIT TO NUMBER OF ITERATIONS
C  TB - BASE TEMPERATURE
C  TE - TIP TEMPERATURE
C  HT - FIN HEIGHT
C  DEL0 - BASE THICKNESS
C  DELE - TIP THICKNESS
C  K - FIN CONDUCTIVITY
C  K1 - CONSTANT = 2.0*SB*EMIS
C  K2 - CONSTANT = E*ABS
C  Q - HEAT INPUT THRU BASE
C  L - FIN LENGTH
C
C----------------------------------------------------------------------
C
      SUBROUTINE MDLIN6(FCN6,X1,X2,XR,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      REAL*8 XERR,FSAVE,F1,F2,FR,X1,X2,XR,XTOL,FTOL,TB,TE,HT,DEL0,
     &       DELE,K,K1,K2,Q,L,FCN6,X1SAV,X2SAV,F1SAV,F2SAV
      INTEGER I,J,PASS
      CHARACTER*2 ANS
      DATA XTOL,FTOL/0.0001D0,0.00001D0/
C
C-------- INITIALIZE VALUES -------------------------------------------
C
      PRINT 200
      X1SAV = X1
      X2SAV = X2
      F1 = FCN6(X1,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      F2 = FCN6(X2,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      F1SAV = F1
      F2SAV = F2
      FSAVE = F1
C
C-------- INITIALIZE COUNTER ------------------------------------------
C
      I = 0
      J = 1
      PASS = 0
C
C-------- LIMIT SEARCH TO INCREMENTS OF 50 ITERATIONS -----------------
C
    5 I = I + 1
```

```
      IF(I.GT.J*50) GOTO 10
      XR = X2-F2*(X2-X1)/(F2-F1)
      FR = FCN6(XR,TB,TE,HT,DEL0,DELE,K,K1,K2,Q,L)
      IF(PASS.EQ.0) PRINT 100,I
      IF(PASS.EQ.1) PRINT 200
      IF(PASS.GE.1) PRINT 300,I,XR,FR
      IF(PASS.GE.1) PASS = PASS + 1
C
C-------- CHECK STOPPING CRITERIA ---------------------------------
C
      XERR = DABS(X1-X2)/2.0D0
      IF(XERR.LE.XTOL.OR.DABS(FR).LE.FTOL) RETURN
C
C-------- FIND NEW POINT --------------------------------------------
C
      IF(FR*F1.GE.0.0D0) THEN
    X1 = XR
    F1 = FR
    IF(FR*FSAVE.GT.0.0D0) F2 = F2/2.0D0
    FSAVE = FR
      ELSE
    X2 = XR
    F2 = FR
    IF(FR*FSAVE.GT.0.0D0) F1 = F1/2.0D0
    FSAVE = FR
      ENDIF
      GOTO 5
C
C-------- FAIL TO FIND ROOT, CONTINUE SEARCH ? ---------------------
C
   10 PRINT 400,J*50
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
        GOTO 10
      ENDIF
      IF(ANS.EQ.'N') GOTO 15
      J = J + 1
      I = I - 1
      GOTO 5
C
C-------- FAIL TO FIND ROOT, LOOK AT FUNTION VALUES ? -----------------
C
   15 PRINT 700
      READ 500,ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'N') THEN
        PRINT 600
     GOTO 15
      ENDIF
      IF(ANS.EQ.'N') STOP
```

```
      X1 = X1SAV
      X2 = X2SAV
      F1 = F1SAV
      F2 = F2SAV
      I = 0
      J = 1
      PASS = 1
      GOTO 5
C
C ------------------------------------------------------------------
C
  100 FORMAT('+', 'ITERATION : ',I3,'                               ')
  200 FORMAT(/,' COMMENCING SEARCH FOR ROOT IN INTERVAL....         ',/)
  300 FORMAT(' AT ITERATION',I3,3X,' X = ',D15.6,3X,' F(X) = ',D15.6)
  400 FORMAT(/,' FAILED TO FIND ROOT AFTER ',I3,' ITERATIONS.'      ,
     & ' CONTINUE SEARCH (Y OR N) ?'                                )
  500 FORMAT(A2)
  600 FORMAT(/' ***************** INCORRECT ANSWER *******************')
  700 FORMAT(/,' WOULD YOU LIKE TO SEE FUNCTION VALUES (Y OR N )?   ',/,
     & ' THIS WILL GIVE SOME IDEA OF WHAT THE FUNCTION IS DOING.    ',/,
     & ' IF F(XR) IS OSCILLATING ABOUT A CERTAIN VALUE OR ITS       ',/,
     & ' ABSOLUTE VALUE IS INCREASING THEN THERE IS LITTLE          ',/,
     & ' PROBABILITY THAT A ROOT WILL BE FOUND.                     ')
      END
C
C8A ***** SI: (TB,TE,L,HT,DEL,DEL0,DELE,K,E,,QI,Q) ******************
C
C   THIS SUBROUTINE CONVERTS ARGUMENTS TO SI UNITS
C
      SUBROUTINE SI(TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q)
      REAL*8 X,TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q
      REAL*8 DEGCF,MFT,WBTU,COND
C
C-------- DEFINE CONVERSION FUNCTIONS ------------------------------
C
C-------- CONVERT TO CENTIGRADE FROM FAHRENHEIT ---------------------
C
      DEGCF(X) = 5.0D0*(X-32.0D0)/9.0D0
C
C-------- CONVERT TO METERS FROM FEET ------------------------------
C
      MFT(X) = X/3.280833D0
C
C-------- CONVERT TO W FROM BTU/HR ---------------------------------
C
      WBTU(X) = X/3.412D0
C
C-------- CONVERT TO W/M-K FROM BTU/HR-FT-F (CONDUCTIVITY) ----------
C
      COND(X) = X*1.729577D0
```

```
C
C-------- CONVERT ELEMENTS ------------------------------------------
C
      TB = DEGCF(TB)
      TE = DEGCF(TE)
      L = MFT(L)
      HT = MFT(HT)
      DEL = MFT(DEL)
      DEL0 = MFT(DEL0)
      DELE = MFT(DELE)
      K = COND(K)
      E = WBTU(E)
      QI = WBTU(QI)
      Q = WBTU(Q)
      RETURN
      END
C
C9A ***** ENG: (TB,TE,L,HT,DEL,DEL0,DELE,K,E,,QI,Q) ******************
C
C   THIS SUBROUTINE CONVERTS ARGUMENTS TO ENG UNITS
C
      SUBROUTINE ENG(TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q)
      REAL*8 X,TB,TE,L,HT,DEL,DEL0,DELE,K,E,QI,Q
      REAL*8 DEGFC,FTM,BTUW,COND
C
C-------- DEFINE CONVERSION FUNCTIONS -------------------------------
C
C-------- CONVERT TO FAHRENHEIT FROM CENTIGRADE----------------------
C
      DEGFC(X) = 9.0D0*X/5.0D0 + 32.0D0
C
C-------- CONVERT TO FEET FROM METERS -------------------------------
C
      FTM(X) = X*3.280833D0
C
C-------- CONVERT TO BTU/HR FROM W ---------------------------------
C
      BTUW(X) = X*3.412D0
C
C-------- CONVERT TO BTU/HR-FT-F FROM W/M-K (CONDUCTIVITY) ----------
C
      COND(X) = X/1.729577D0
C
C-------- CONVERT ELEMENTS ------------------------------------------
C
      TB = DEGFC(TB)
      TE = DEGFC(TE)
      L = FTM(L)
      HT = FTM(HT)
      DEL = FTM(DEL)
```

```
      DEL0 = FTM(DEL0)
      DELE = FTM(DELE)
      K = COND(K)
      E = BTUW(E)
      QI = BTUW(QI)
      Q = BTUW(Q)
      RETURN
      END
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
```

LIST OF REFERENCES

1. Morris, W., The American Heritage Dictionary of the English Language, American Heritage Publishing Company, Incorporated, 1970.

2. Yamaha, XS400 H/SH Service Manual, Yamaha Motor Corporation, 1981.

3. Office of the Space Station National Aeronautics and Space Administration, "Space Station Freedom Capital Development Plan Fiscal Year 1990", NASA, 1989.

4. DeWitt, D. P. and Incropera, F. P., Fundamentals of Heat Transfer, John Wiley and Sons, 1981.

5. Ganic, E. N., Hartnett, J. P. and Rohsenow, W. M., Handbook of Heat Transfer Applications, McGraw-Hill Book Company, 1985.

6. Howell, J. R., A Catalog of Radiation Configuration Factors, McGraw-Hill Book Company, 1982.

7. Haberman, R., Elementary Applied Partial Differential Equations with Fourier Series and Boundary Value Problems, Prentice-Hall, Incorporated, 1987.

8. Bohn, M. S. and Kreith, F., Principles of Heat Transfer, Harper and Row Publishers, 1986.

9. Agrawal, B. N., Design of Geosysnchronous Spacecraft, Prentice-Hall Incorporated, 1986.

10. Kraus, A. D., "Class Notes for Thermal Control of Space Craft (AE 3804) for Spring Quarter", 1989.

11. Ferziger, J. H., Numerical Methods for Engineering Application, John Wiley & Sons, 1981.

12. Gerald, C. F. and Wheatley, P. O., Applied Numerical Analysis, Addison-Wesley Publishing Company, 1989.

13. Conte, S. D. and deBoor, C., Elementary Numerical Analysis, McGraw-Hill Book Company, 1972.

14. Hamming, R. W., Introduction to Applied Numerical Analysis, McGraw-Hill Book Company, 1971.

15. Page-Jones, M., The Practical Guide to Structured Systems Design, Yourdon Press, 1988.

16. Kern, D.Q. and A.D. Kraus, Extended Surface Heat Transfer, McGraw-Hill Book Company, 1972.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Superintendent Attn: Library, Code 1424 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Naval Postgraduate School Attn: Prof. Allan D. Kraus, Code EC/Ks Monterey, California 93943-5000	1
4.	Naval Postgraduate School Attn: Prof. Sue Brown, Code MA Br Monterey, California 93943-5000	1
5.	Naval Postgraduate School Attn: Curriculum Office, Code 39 Monterey, California 93943-5000	1
6.	Naval Postgraduate School Attn: Chairman, Code AA Monterey, California 93943-5000	1